

Trabajo Fin de Grado
Ingeniería de las Tecnologías Industriales
Mención en Automática

Gestión activa de la demanda eléctrica mediante
sistema domótico

Autor: Ezequiel Bravo del Cid

Tutores: Jose María Maestre Torreblanca

Jesús Iván Maza Alcañiz

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Gestión activa de la demanda eléctrica mediante sistema domótico

Autor:
Ezequiel Bravo del Cid

Tutores:
Jose María Maestre Torreblanca
Profesor catedrático
Jesús Ivan Maza Alcañiz
Profesor titular

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2021

Trabajo Fin de Grado: Gestión activa de la demanda eléctrica mediante sistema domótico

Autor: Ezequiel Bravo del Cid

Tutores: Jose María Maestre Torreblanca
Jesús Iván Maza Alcañiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

*A mi abuelo, que en paz
descanse.*

Agradecimientos

A mi familia, por dármele todo en esta vida sin recibir nada a cambio. Porque el amor de los padres es incondicional y nada en esta vida se le puede asemejar. Gracias por la educación que me habéis dado y los valores que me habéis transmitido. Sé que nunca podré agradeceréoslo lo suficiente.

A mis amigos de la Universidad, que me han acompañado estos cuatro años de carrera, con los que he compartido muchas experiencias, tanto buenas como malas. Gracias por ser de las mejores personas que conozco. En especial, agradecer a Jesús Arteaga, por acompañarme en esta etapa desde el primer día en aquella clase de Física I, y a Alejandro Pineda, por abrirme tantas puertas, por presentarme a tanta gente maravillosa, por su cariño y por su confianza en mí.

A mis amigos de Madrid, en especial a Koke, Carlota e Itziar, que, a pesar de la distancia, han sido un apoyo muy importante para mí a lo largo de la carrera. Da igual el tiempo que pase, se que ellos estarán ahí para lo que necesite y no tengo palabras para describir lo afortunado que me siento de haberles conocido.

A mi psicóloga, Alicia Rodríguez Aguilar, por escucharme siempre, por ayudarme a intentar ser siempre la mejor versión de mi mismo, por enseñarme la importancia de la salud mental y cómo cuidarla.

A mis tutores, Iván Maza y Jose María Maestre, por su gran disposición y ayuda a lo largo de la realización de este proyecto.

Por último, a mí mismo, por lo necesario que se hace realmente creer en uno mismo en esta vida. Por trabajar duro, por tener espíritu de sacrificio y ese constante afán de superación personal en todos los aspectos de la vida, tanto personales como académicos.

Sin ninguna de estas personas yo no sería la persona que actualmente soy y no me queda nada más que agradacersele a todos. Todas estas personas tendrán un lugar en mi corazón para el resto de mi vida.

Ezequiel Bravo del Cid

Sevilla, 2021

Resumen

El objetivo de este proyecto es demostrar mediante ensayos y experimentos el ahorro energético que se puede conseguir en una vivienda gracias a la gestión activa de la energía eléctrica. Para ello, se utilizarán dos tecnologías de control diferentes: Control Predictivo de Modelo (MPC) y control basado en reglas. Se realizarán diferentes simulaciones y ensayos cuyos resultados serán comparados. Para llevar a cabo las simulaciones, se utilizarán los programas de Matlab y Home I/O simultáneamente. El proyecto se centrará sobre todo en el control de los calefactores presentes en la vivienda simulada de Home I/O, ya que presentan el mayor impacto en el consumo eléctrico. Como fuente de alimentación de energía eléctrica a la vivienda, habrá dos posibles casos: el primero, en el que la vivienda solamente pueda recibir energía eléctrica de la red eléctrica pública; y el segundo, en el que la vivienda dispone de una microred que posee diversas fuentes de generación de energía eléctrica, permitiendo la optimización del proceso de elección de que fuente utilizar en cada instante de tiempo.

Abstract

The aim of this project is to demonstrate through tests and experiments the energy savings that can be achieved in a home thanks to the active management of electrical energy. For this, two different control technologies will be used: Model Predictive Control (MPC) and rule-based control. Different simulations and tests will be carried out, the results of which will be compared. To carry out the simulations, the Matlab and Home I/O programs will be used simultaneously. The project will focus above all on the control of the heaters present in the simulated home of Home I/O, since they have the greatest impact on electricity consumption. As a source of electrical energy supply to the home, there will be two possible cases: the first, in which the home can only receive electrical energy from the public electricity grid; and the second, in which the home has a microgrid that has various sources of electricity generation, allowing the optimization of the process of choosing which source to use at any moment in time.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
1 Introducción	1
1.1 Contexto actual y alcance del proyecto	1
1.2 Herramientas e ideas para la realización del proyecto	2
2 Home I/O y la Microred	5
2.1 Planos de la Vivienda y consumos eléctricos	5
2.1.1 Dispositivos de iluminación	6
2.1.2 Interruptores de la iluminación y sensores de luminosidad	7
2.1.3 Persianas y sus correspondientes interruptores	7
2.1.4 Calefactores y termostatos	8
2.1.5 Detectores de movimiento, humo, infrarrojos y de apertura de puerta	9
2.1.6 Otros dispositivos	9
2.2 Comunicación HOME I/O - Matlab	10
2.2.1 Modos de funcionamiento de los dispositivos de la vivienda	10
2.2.2 Direcciones de memoria de los dispositivos	10
2.2.3 Lectura/escritura de variables de Home I/O utilizando Matlab	11
2.3 Microred simulada en Simulink	15
2.3.1 Componentes de la microred	15
2.3.2 Funcionamiento de la microred	16
2.3.3 Modelo lineal de la microred y restricciones	16
2.3.4 Controlador MPC de la microred	18
3 Modelos y Controladores	21
3.1 Control predictivo basado en modelo (calefactor + microred) para el Sistema A	21
3.1.1 Diseño del modelo	22
3.1.2 Validación del modelo	22
3.1.3 Unión del modelo lineal de la microred con el modelo del calefactor	26
3.1.4 Implementación del controlador y del modelo en Simulink	27
3.2 Control basado en reglas para el Sistema B	29
3.2.1 Lectura de las temperaturas de las habitaciones	29
3.2.2 Asignación de referencias de temperatura para cada habitación	30
3.2.3 Controladores on/off con histéresis	31
3.2.4 Modulación automática de los calefactores	32
3.3 Control predictivo basado en modelo (solo microred) para el Sistema A	37
4 Análisis de estudios	40
4.1 Análisis y resultados del Sistema A	40

4.1.1	Ensayos utilizando el controlador MPC	40
4.2	<i>Análisis y resultados del Sistema B</i>	44
4.2.1	Ensayos con potencia máxima constante	44
4.2.2	Ensayo con el número de calefactores activados constante	46
4.2.3	Ensayo con ninguna variable constante	47
5	CONCLUSIONES Y TRABAJOS FUTUROS	49
5.1	<i>Limitaciones encontradas</i>	49
5.2	<i>Conclusiones</i>	49
5.3	<i>Trabajos futuros</i>	49
	Referencias	51
	ANEXOS	53

ÍNDICE DE TABLAS

Tabla 2-1. Potencias eléctricas consumidas por las distintas luminarias de la vivienda	7
Tabla 2-2. Potencias eléctricas consumidas por los calefactores de la vivienda.	8
Tabla 3-1. Tabla de transición de estados de los controladores on/off con histeresis	32

ÍNDICE DE FIGURAS

Figura 1-1. Precio medio del euro por MWh en España 2010-2021 [7]	1
Figura 1-2. Precio medio diario del euro por MWh en el mercado mayorista en España en 2021 [8]	2
Figura 2-1. Interfaz visual de HOME I/O	5
Figura 2-2. Plano de distribución de las luminarias de la vivienda	6
Figura 2-3. Plano de distribución de los interruptores de la iluminación y sensores de luminosidad de la vivienda	7
Figura 2-4. Plano de distribución de las persianas de la vivienda con sus respectivos interruptores.	7
Figura 2-5. Plano de distribución de los calefactores de la vivienda con sus correspondientes termostatos	8
Figura 2-6. Plano de la distribución de los diferentes detectores de la vivienda	9
Figura 2-7. Plano de distribución de otros dispositivos automatizables de la vivienda	9
Figura 2-8. Interfaz visual de la herramienta “Engine I/O Explorer”	11
Figura 2-9. Bloques en Simulink que ejecutan “read.m” y “write.m”	11
Figura 2-10. Función setup de la función de lectura de variables de Home I/O	12
Figura 2-11. Función output de la función de lectura de variables de Home I/O	12
Figura 2-12. Función setup de la función de escritura de variables de Home I/O	13
Figura 2-13. Función output de la función de escritura de variables de Home I/O	14
Figura 2-14. Bloque de Simulink RealTime_Pacer para la simulación en tiempo real	14
Figura 2-15. Microred simulada en Simulink	15
Figura 2-16. Código para la creación del controlador MPC de la microred	18
Figura 2-17. Microred definitiva simulada en Simulink con su respectivo controlador MPC	19
Figura 3-1. Habitación de los niños (zona H)	21
Figura 3-2. Datos obtenidos del experimento 1 para la validación del modelo	23
Figura 3-3. Comparación de respuestas simuladas del experimento 1 y el modelo	23
Figura 3-4. Datos obtenidos del experimento 2 para la validación del modelo	24
Figura 3-5. Comparación de respuestas simuladas del experimento 2 y el modelo	24
Figura 3-6. Datos obtenidos del experimento 3 para la validación del modelo	25
Figura 3-7. Comparación de respuestas simuladas del experimento 3 y el modelo	25
Figura 3-8. Código para la creación del controlador MPC del modelo de la microred + calefactor	27
Figura 3-9. Modelo en Simulink del controlador MPC de la microred+calefactor	28
Figura 3-10. Modelo en Simulink del Sistema A con el controlador MPC completo (modelo microred+calefactor)	28
Figura 3-11. Bloque de lectura de las temperaturas de todas las habitaciones en Simulink	29
Figura 3-12. Asignación de referencia de temperatura para cada habitación en Simulink	30
Figura 3-13. Controlador on/off con histeresis para controlar la temperatura de cada habitación en Simulink	31
Figura 3-14. Funcion setup de la function “modulado_y_apagado.m”	32
Figura 3-15. Definición de entradas en la función output de la function “modulado_y_apagado.m”	33

Figura 3-16. Algoritmo secuencial de apagado en la función output de la función “modulado_y_apagado.m”	34
Figura 3-17. Algoritmo secuencial de reencendido en la función output de la función “modulado_y_apagado.m”	34
Figura 3-18. Definición de salidas en la función output de la función “modulado_y_apagado.m”	35
Figura 3-19. Bloque de Simulink que ejecuta la función “modulación_y_apagado.m”	36
Figura 3-20. Bloque de escritura de las tensiones a todos los calefactores en Simulink	37
Figura 3-21. Modelo en Simulink del Sistema A con el controlador MPC (solo microred)	38
Figura 4-1. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) n°1 sistema A	40
Figura 4-2. Evolución temporal de las variables del calefactor, ensayo MPC (solo microred) n°1 sistema A	41
Figura 4-3. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) n°2 sistema A	41
Figura 4-4. Evolución temporal de las variables del calefactor, ensayo MPC (solo microred) n°2 sistema A	42
Figura 4-5. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) n°3 sistema A	43
Figura 4-6. Evolución temporal de las variables del calefactor, ensayo MPC (solo microred) n°3 sistema A	44
Figura 4-7. Ensayo sistema B con potencia máxima constante encendiendo progresivamente los calefactores	45
Figura 4-8. Ensayo sistema B con potencia máxima constante apagando progresivamente los calefactores	46
Figura 4-9. Ensayo sistema B con número de calefactores activados constante	47
Figura 4-10. Ensayo sistema B con ninguna variable constante	48
Figura 5-1. Horarios de las distintas partes que tiene la tarifa de luz con discriminación horaria [9]	50

1 INTRODUCCIÓN

La sociedad actual cada vez es más dependiente de la energía eléctrica. Es un recurso que todas las personas usan en su rutina diaria y que sin la cual, sería difícil actualmente contemplar una vida sin ella. Con el paso de los años, esto se acentuará cada vez más, debido a que con el cambio climático y la gran contaminación que existe en el mundo, se reducirá cada vez más el uso de combustibles fósiles y se reemplazarán por el uso de la energía eléctrica. Ésta se puede obtener de forma limpia a través de las energías renovables y no produce gases de contaminación atmosférica.

En este capítulo, se va a tratar el contexto actual en el que se encuentra la sociedad y por qué es realmente necesario hacer una gestión adecuada de la energía eléctrica, y las herramientas que tenemos para llevar a cabo esta gestión de forma automática o en la que participe parcialmente el usuario.

1.1 Contexto actual y alcance del proyecto

En Diciembre de 2019, apareció en China un nuevo coronavirus, que todavía en nuestros días, en pleno 2021, sigue causando estragos en todo el mundo. Este nuevo coronavirus, denominado coronavirus SARS-CoV-2, ha originado una pandemia mundial de la que todavía el mundo no ha conseguido salir. Bien es cierto, gracias a la investigación y desarrollo de las vacunas, que el impacto que tiene este virus en la sociedad cada vez es menor por la demostrada efectividad de dichas vacunas.

Esta pandemia mundial ha originado una gran crisis económica mundial, teniendo un impacto negativo inevitable en la vida de todas las personas. Entre estas consecuencias, se va a destacar el precio de la luz en España, que actualmente se encuentra en precios de máximos históricos. No es solo debido a la pandemia, existen otros motivos más detrás de este suceso, pero estos están fuera del alcance de este proyecto.

El precio medio final de la electricidad en España se ha visto incrementado radicalmente este último año. Se puede observar este hecho en el siguiente gráfico.

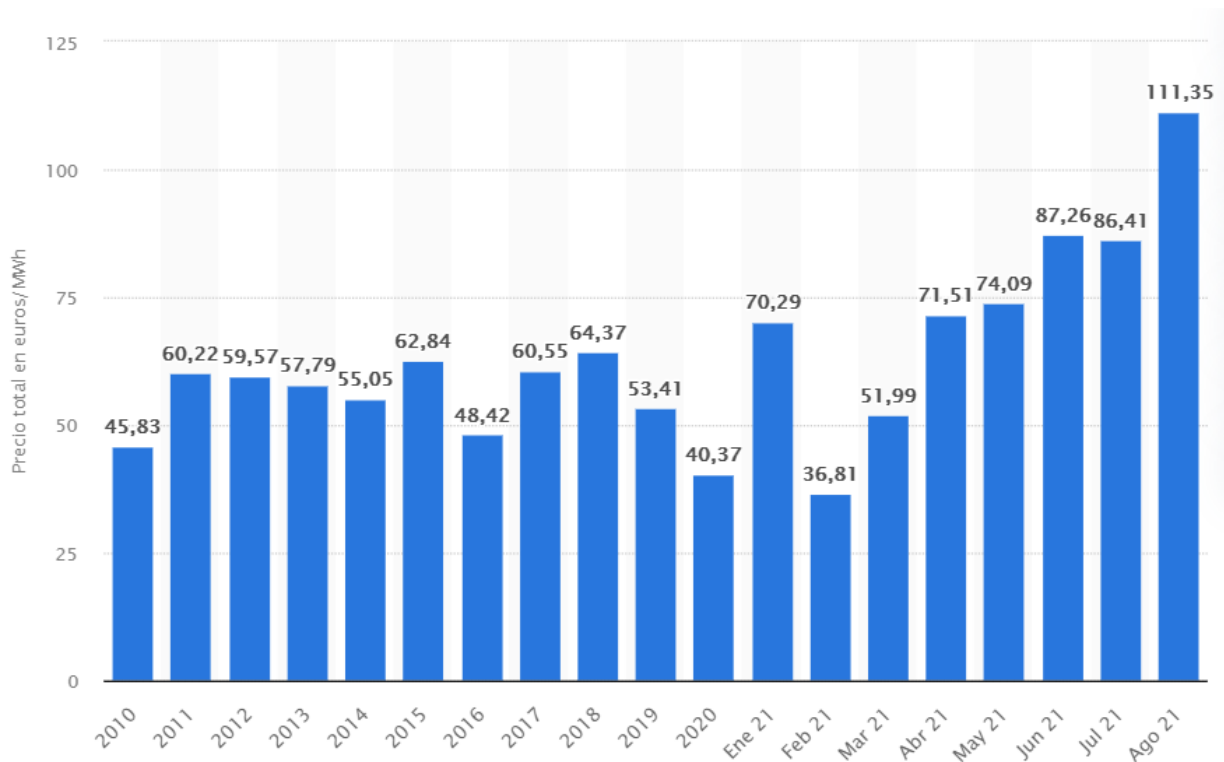


Figura 1-1. Precio medio del euro por MWh en España 2010-2021 [7]

Pero este precio de la electricidad no ha terminado su acenso todavía. En el siguiente gráfico, se observa más detalladamente la evolución del precio del megavatio hora a lo largo de este año.



Figura 1-2. Precio medio diario del euro por MWh en el mercado mayorista en España en 2021 [8]

Hoy en día, como se puede apreciar, sigue el precio al alza y cada día que pasa se van alcanzando nuevos máximos históricos. Se desconoce cual es el precio máximo que se llegará a alcanzar, ni cuando lo hará. Por este motivo, se hace de vital importancia una gestión adecuada de la energía eléctrica más que nunca, ya que una gestión mediocre de la electricidad ahora mismo puede provocar una subida de la factura de la luz considerable.

Por tanto, el objetivo de este proyecto es dar a conocer ideas o formas para conseguir una gestión adecuada de la electricidad a nivel personal, en la vivienda de cada individuo. Igualmente, estas ideas se podrían aplicar a instalaciones más grandes, ya sean industrias, grandes edificios...

A continuación, se hará una breve introducción de las herramientas que se van a utilizar para este objetivo y detallar brevemente las diferentes formas de conseguir una mejor gestión de la electricidad con la tecnología actual.

1.2 Herramientas e ideas para la realización del proyecto

Para llevar a cabo la realización de este proyecto, se va a hacer uso de dos programas: Matlab y Home I/O.

Matlab es una plataforma de programación y cálculo numérico cuyo uso está muy extendido en el mundo de la ingeniería. Es el programa que nos va a permitir ejercer el control sobre los aparatos eléctricos de la vivienda de forma automática.

Home I/O es un entorno de simulación interactivo que permite el control sobre una vivienda totalmente automatizable. Esta vivienda tiene diferentes aparatos eléctricos, que se pueden controlar utilizando programas externos.

La forma de funcionar a lo largo del proyecto será la siguiente:

- Se leen variables de interés de la vivienda, ya sean las temperaturas de las habitaciones, luminosidad, detectores...

- Esa información la introducimos en Matlab Simulink y se tomarán una serie de decisiones.
- Una vez tomadas, escribiremos sobre las variables de Home I/O para llevar a cabo las acciones que han sido estimadas oportunas, ya sea activar o desactivar algún aparato eléctrico, disminuir su potencia, aumentarla...

En cuanto al ahorro de la energía eléctrica, se plantean dos situaciones en las que se aporta una solución diferente:

- Caso 1: la vivienda solo dispone de la red eléctrica pública para su alimentación de energía eléctrica. En este caso, puede resultar útil el control basado en reglas. Esta técnica de control consiste en reducir la potencia o apagar los aparatos eléctricos en un orden establecido. Este orden de prioridad sería establecido por el usuario, y el algoritmo lo ejecutaría. Esto es muy interesante, sobre todo, en la situación actual en la que hay diferentes tarifas de electricidad según la hora del día. Se podría limitar la potencia de la vivienda a un valor más alto en aquellas horas que el precio de la electricidad sea menor, y a un valor más bajo en aquellas horas que el precio de la electricidad sea mayor. El objetivo es que, aunque se activen más aparatos eléctricos de la casa, la potencia que consume la vivienda no sea mayor que la potencia límite. Para ello, se reduce la potencia de los aparatos eléctricos según un orden de prioridad establecido por el usuario.
- Caso 2: la vivienda dispone de una microred en la que hay diversas fuentes de generación de energía eléctrica que se pueden utilizar. En este caso, usar un controlador predictivo por modelo (MPC) sería muy productivo. Para la implementación de este controlador, primero es necesario tener un modelo del sistema que se quiere controlar. El modelo tendrá ciertas variables de entrada, de salida y perturbaciones. Una vez se tiene identificado el sistema, se puede programar el controlador MPC. Como en cualquier tipo de controlador en bucle cerrado, se introducirá una referencia para cada salida. El objetivo del controlador es determinar los valores óptimos que deberían tomar las variables de entrada para llevar las diferentes salidas a sus respectivas referencias. El criterio para este objetivo tiene lugar en función de los pesos. Los pesos son valores adimensionales que se dan a las variables de entrada, a su incremento, y a las variables de salida. En función de estos pesos, los valores óptimos de las variables de entrada serán diferentes. Estos pesos influyen debido a que se utilizan en la función de coste. Si se aumenta por ejemplo drásticamente el peso de una variable de entrada, se penalizará más su uso y el controlador entenderá que se prefiere que esa variable se utilice lo menos posible. Otras peculiaridades de este controlador es el hecho de que tiene la capacidad de anticipar acontecimientos futuros y tomar acciones de control consecuentemente. Se encarga de optimizar las variables en un horizonte de tiempo finito. También permite establecer restricciones para las diferentes variables de entrada, salida e incrementos de las variables de entrada.

Estos controladores son útiles también a la hora de gestionar el consumo eléctrico en una vivienda, estableciendo una relación entre la potencia y su impacto en la variable que queramos controlar de la habitación, ya sea la temperatura, luminosidad... La entrada sería la tensión o potencia a la que trabaja el aparato eléctrico, y la salida la variable que queremos controlar de la habitación o estancia. En este caso, para alimentar a la vivienda se usará una microred simulada en Matlab Simulink.

Esta microred tiene diferentes fuentes de alimentación para generar potencia. Se programará un controlador MPC también que se encargará de estimar cuanto debe usar de cada una en cada momento, en función de la demanda de la carga. Así, podríamos aparte de controlar la potencia del aparato eléctrico de forma optimizada, minimizar a su vez la cantidad de potencia que se absorbe de la red eléctrica pública utilizando el mismo controlador.

2 HOME I/O Y LA MICRORED

Home I/O es el programa utilizado para la simulación de la automatización de la vivienda. El programa se utiliza como si fuera un videojuego, en el que el usuario puede desplazarse dentro de la vivienda, pudiendo interactuar con los diferentes sensores y actuadores disponibles en la vivienda y en su entorno. En la siguiente figura, se puede observar una captura realizada de lo que ve el usuario dentro del programa:



Figura 2-1. Interfaz visual de HOME I/O

Como se puede apreciar, se obtiene una continua información de interés relativa a la vivienda y al tiempo, como puede ser la temperatura actual en el exterior, la temperatura máxima y mínima del día en concreto en el que nos encontremos, la humedad, el viento, la nubosidad... Esto es importante conocerlo, ya que afectará a la temperatura que tengamos en el interior de la casa. También destacar que en la esquina inferior derecha aparece información relativa al consumo eléctrico: potencia instantánea consumida, energía consumida a lo largo del día, energía consumida a lo largo del mes...

2.1 Planos de la Vivienda y consumos eléctricos

A continuación, se van a mostrar los diferentes planos de la vivienda para localizar los diferentes dispositivos a automatizar que existen en ella y sus respectivos consumos eléctricos. [1]

2.1.1 Dispositivos de iluminación

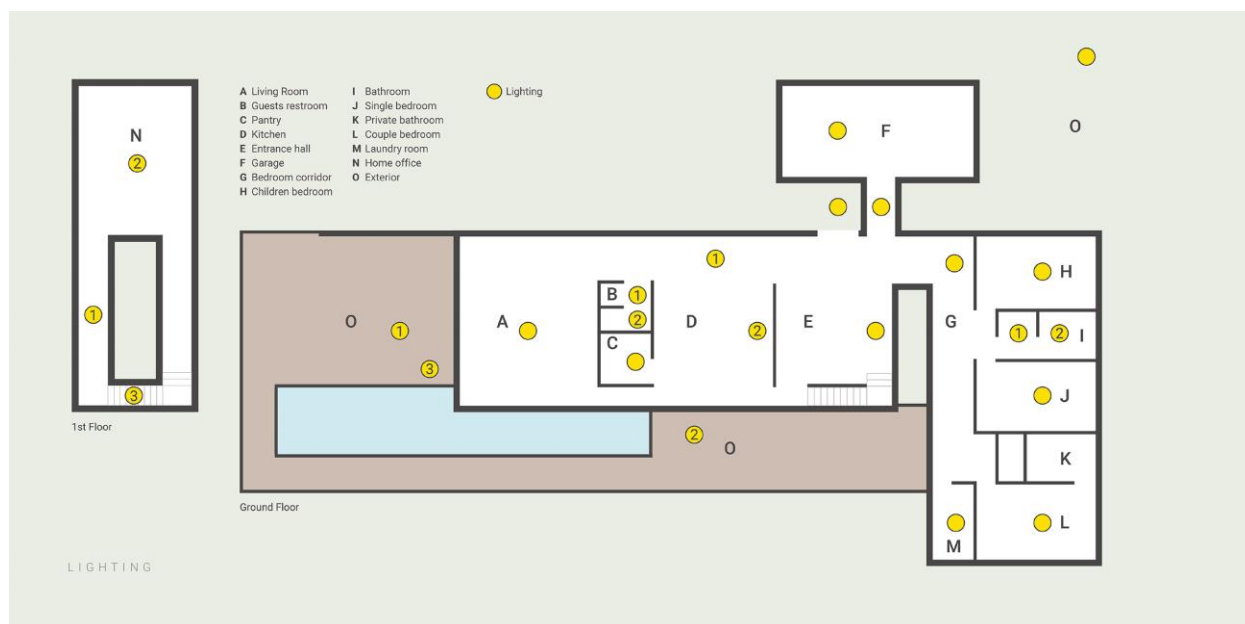


Figura 2-2. Plano de distribución de las luminarias de la vivienda

En cuanto a la potencia eléctrica que consume cada luminaria:

Habitación/zona	Descripción	Set	Potencia (W)
A	Sala de estar	N/A	60
B	Baño de invitados	Luces 1 / 2	10
C	Despensa	N/A	10
D	Cocina	Luces 1 Luces 2	60 30
E	Hall de entrada	N/A	60
F	Garaje	Luces 1 Luces 2	10 120
G	Pasillo del dormitorio	N/A	50
H	Dormitorio de los niños	N/A	40
I	Baño	Luces 1 Luces 2	10 10
J	Habitación individual	N/A	40
K	Baño privado	N/A	20
L	Habitación para pareja	N/A	40
M	Lavandería	N/A	10
N	Oficina	Luces 1 Luces 2 Luces 3	90 30 30
O	Exterior	Luces porche 1 Luces porche 2 Luces piscina Luces jardín	60 80 160 40

Luces entrada	10
---------------	----

Tabla 2-1. Potencias eléctricas consumidas por las distintas luminarias de la vivienda

2.1.2 Interruptores de la iluminación y sensores de luminosidad

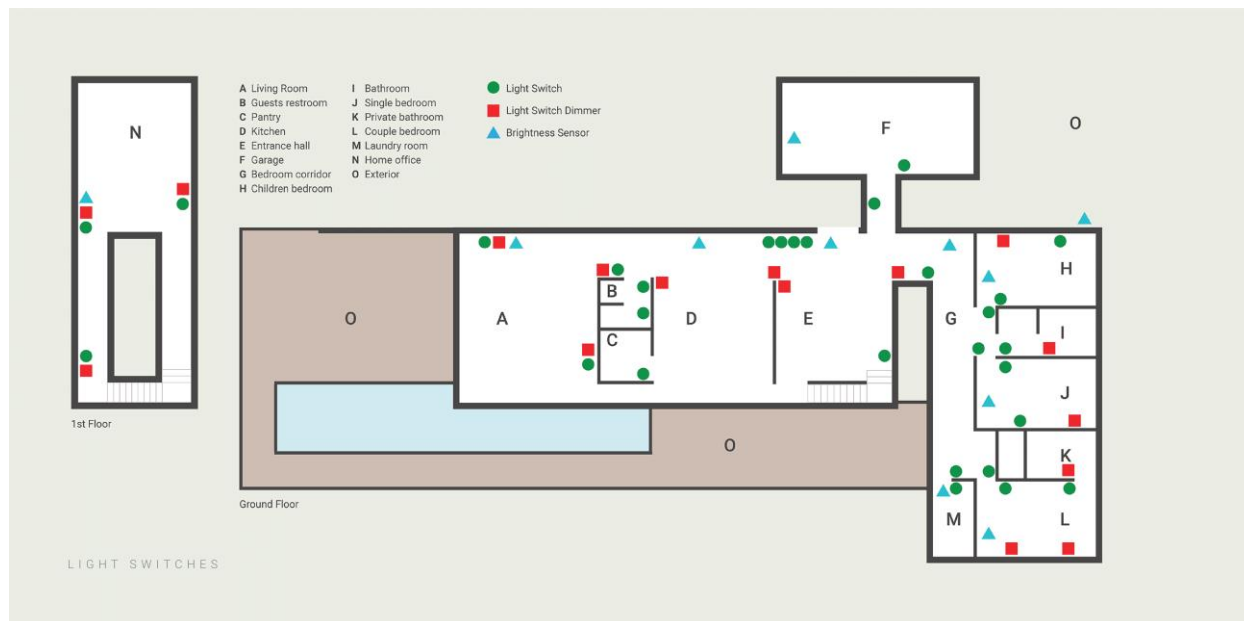


Figura 2-3. Plano de distribución de los interruptores de la iluminación y sensores de luminosidad de la vivienda

En este caso, los interruptores de las luminarias no consumen electricidad y el consumo eléctrico de los sensores de luminosidad es despreciable.

2.1.3 Persianas y sus correspondientes interruptores

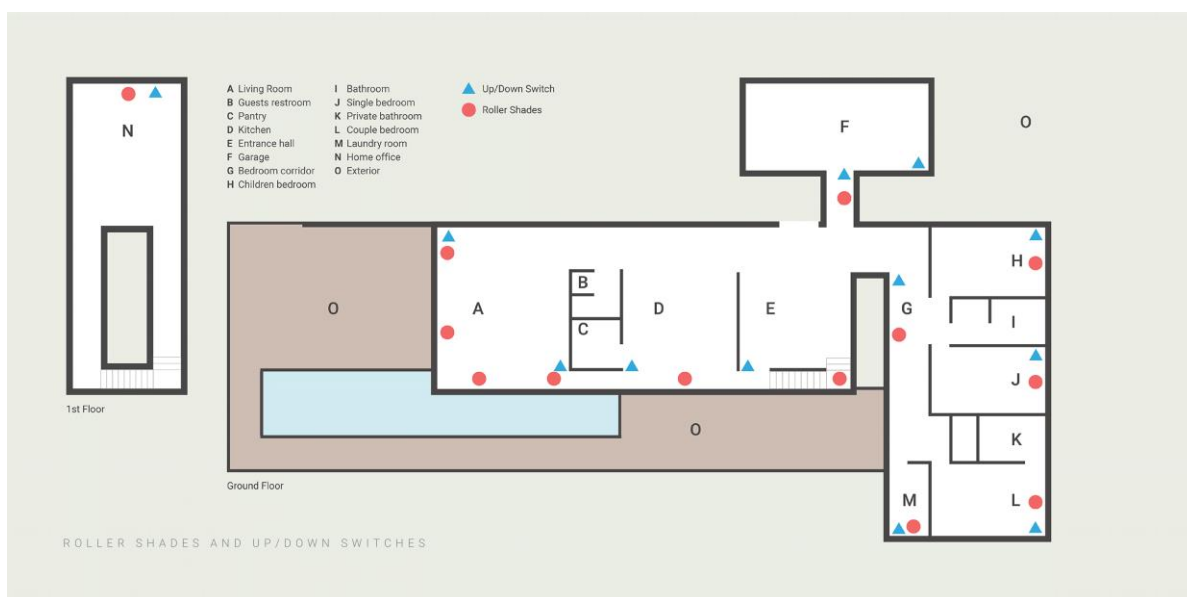


Figura 2-4. Plano de distribución de las persianas de la vivienda con sus respectivos interruptores.

Las persianas tienen cada una un consumo eléctrico de **10 W**.

2.1.4 Calefactores y termostatos

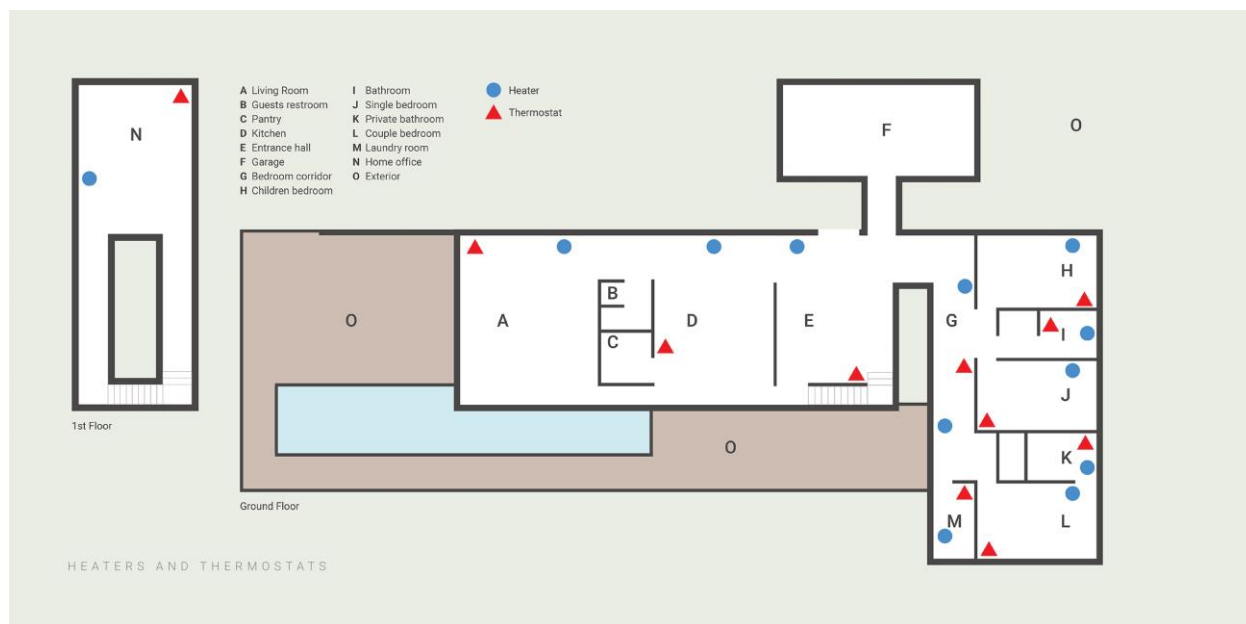


Figura 2-5. Plano de distribución de los calefactores de la vivienda con sus correspondientes termostatos

En cuanto al consumo eléctrico de cada calefactor:

Habitación/zona	Descripción	Potencia (W)
A	Sala de estar	2000
D	Cocina	1500
E	Hall de entrada	1750
G	Pasillo del dormitorio	Calefactor 1: 750 Calefactor 2: 750
H	Dormitorio de los niños	1000
I	Baño	750
J	Habitación individual	750
K	Baño privado	1000
L	Habitación doble	750
M	Lavandería	500
N	Oficina	1500

Tabla 2-2. Potencias eléctricas consumidas por los calefactores de la vivienda.

2.1.5 Detectores de movimiento, humo, infrarrojos y de apertura de puerta

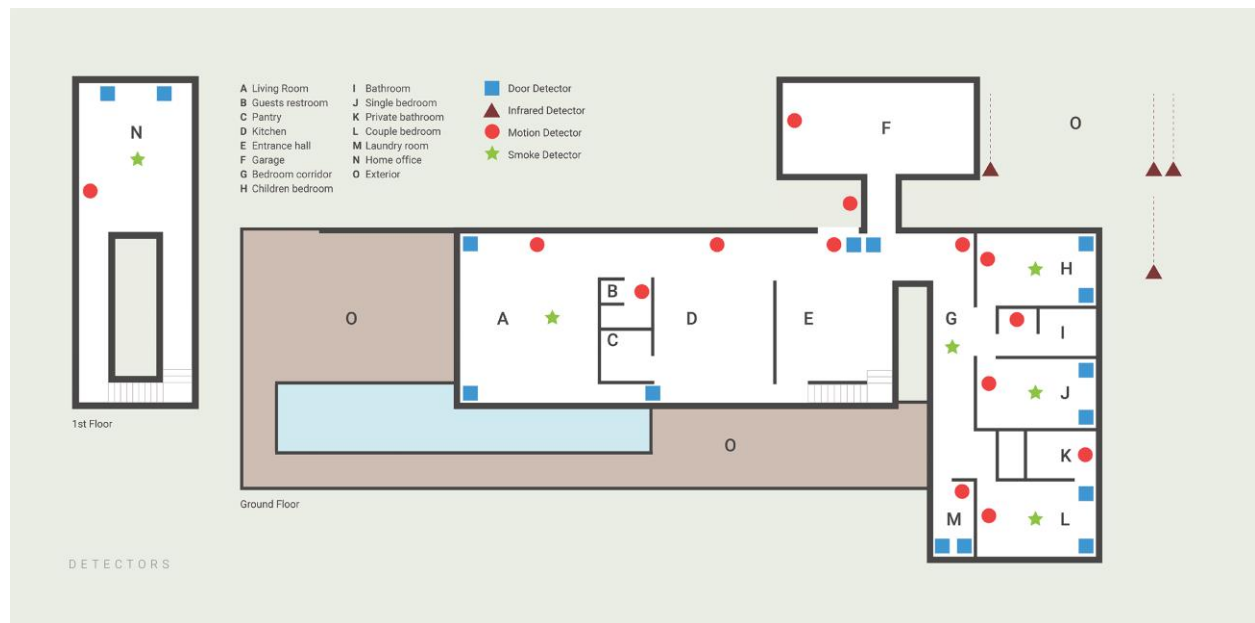


Figura 2-6. Plano de la distribución de los diferentes detectores de la vivienda

En cuanto al consumo eléctrico de los detectores, éste es despreciable.

2.1.6 Otros dispositivos

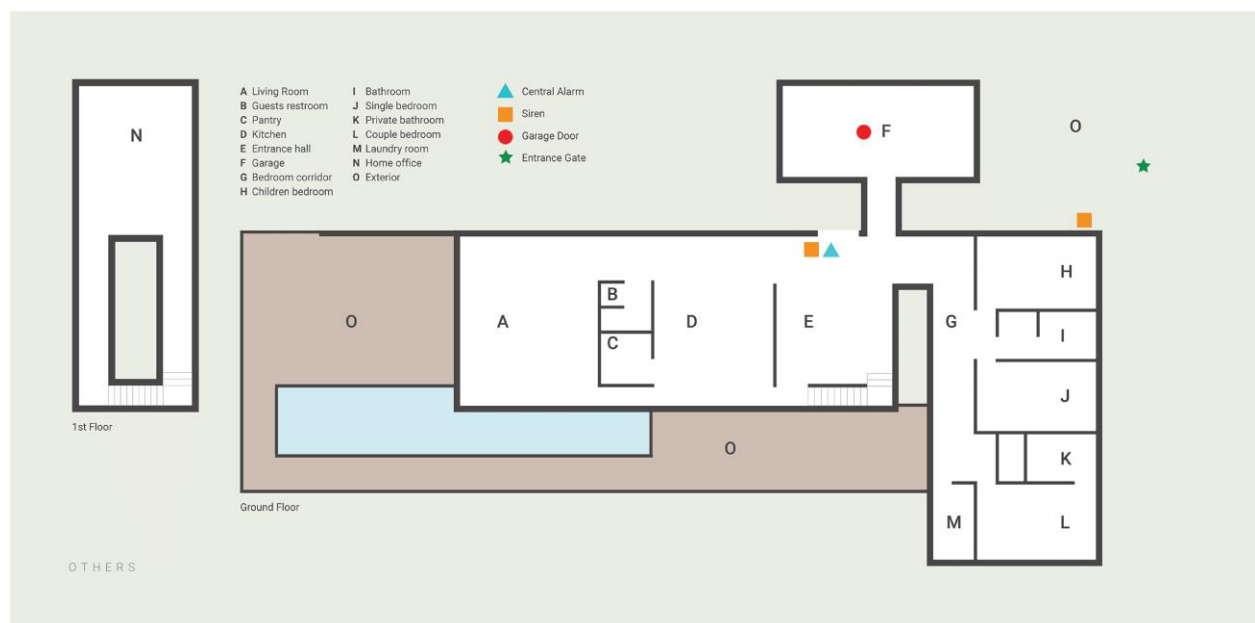


Figura 2-7. Plano de distribución de otros dispositivos automatizables de la vivienda

Estos otros dispositivos son: una alarma central con su sirena que consume una potencia de 30 W, una puerta de garaje de 1000 W y la puerta de entrada al recinto de 1000 W.

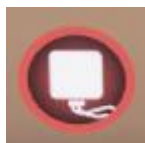
2.2 Comunicación HOME I/O - Matlab

Para la programación y automatización de la vivienda, se va a utilizar el programa de Matlab, ya que Home I/O permite el control y utilización de los dispositivos automatizables mediante el uso de programas externos, como puede ser Matlab.

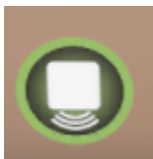
2.2.1 Modos de funcionamiento de los dispositivos de la vivienda

Para poder conectar los dispositivos a programas externos, se deben modificar estos para este propósito. Los dispositivos tienen tres modos de funcionamiento:

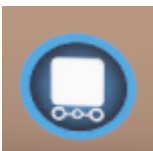
- **Wired mode.** El dispositivo no se puede automatizar. Hace referencia al funcionamiento que tendría en una instalación eléctrica estandar donde todos los dispositivos estan conectados mediante cables. El símbolo es el siguiente:



- **Wireless mode.** El dispositivo se puede controlar mediante la consola del programa. Los dispositivos trabajan con un controlador central inalámbrico que puede programarse para automatizar los dispositivos que se encuentren en este modo de funcionamiento. El símbolo es el siguiente:



- **External mode.** Permite a los dispositivos de entrada/salida ser controlados por programas externos. Su símbolo es el siguiente:



Por tanto, una vez vistos los tres modos de funcionamiento diferentes, el que mayor relevancia tiene para este trabajo es el “external mode”. Todos los dispositivos serán controlados desde Matlab, en concreto, utilizando la herramienta de Simulink.

Para modificar voluntariamente los dispositivos que se quieren controlar, se tiene que crear una partida nueva dentro de Home I/O. Entonces, se recorre la vivienda, buscando los diferentes dispositivos que se quieren controlar. Una vez se encuentren, haciendo click sobre el símbolo mostrado cambiará entre los distintos modos de funcionamiento hasta que se obtenga el deseado en cada uno de los dispositivos.

2.2.2 Direcciones de memoria de los dispositivos

Los dispositivos que se encuentren en “external mode”, tendrán su información disponible para ser accedida a través de programas externos. Las direcciones de memoria de todos los dispositivos (dispositivos de entrada, de salida y variables de memoria) vienen recogidas en los anexos A, B y C, que se encuentran al final de este documento.

Para facilitar el acceso a esta información, existe una herramienta para Home I/O que se denomina “Engine I/O Explorer”, cuyo propósito es mostrar al usuario todos los dispositivos que se encuentren en “external mode”, navegando en los diferentes tipos de variables: inputs, outputs y memories; y a su vez, para cada tipo de variable, recorriendo los diferentes tipos de datos (bit, float, int...). La herramienta también da información de la dirección de memoria que le corresponde a ese dispositivo, el tipo de dato que almacena y su correspondiente valor.

Bit	Byte	Short	Int	Long	Float	Double	String	DateTime	TimeSpan
INPUTS									
Address	Name	Value							
54		0							
55		0							
56		0							
57	H - Brightness Sensor (Analog)	0,08495793							
58	H - Thermostat (Room Temperature)	18,5488262							
59	H - Thermostat (Set Point)	0							
60	H - Roller Shades (Openness)	0							
61		0							
62		0							
OUTPUTS									
Address	Name	Value							
0		<input type="checkbox"/>							
1		<input type="checkbox"/>							
2		<input type="checkbox"/>							
3		<input type="checkbox"/>							
4		<input type="checkbox"/>							
5		<input type="checkbox"/>							
6		<input type="checkbox"/>							
7		<input type="checkbox"/>							
8		<input type="checkbox"/>							
MEMORIES									
Address	Name	Value							
0		<input type="checkbox"/>							
1		<input type="checkbox"/>							
2		<input type="checkbox"/>							
3		<input type="checkbox"/>							
4		<input type="checkbox"/>							
5		<input type="checkbox"/>							
6		<input type="checkbox"/>							
7		<input type="checkbox"/>							
8		<input type="checkbox"/>							
9		<input type="checkbox"/>							
10		<input type="checkbox"/>							
11		<input type="checkbox"/>							

Figura 2-8. Interfaz visual de la herramienta “Engine I/O Explorer”

En la figura anterior se muestra un ejemplo donde se puede observar como el sensor de luminosidad, el termostato y el sensor de apertura de persianas han sido configurados como “external mode”. Son dispositivos de entrada cuyo valor podemos leer y actuar en consecuencia utilizando Matlab o cualquier otro programa externo que lo permita.

2.2.3 Lectura/escritura de variables de Home I/O utilizando Matlab

El objetivo de la utilización de Matlab es leer el valor de variables de entrada de Home I/O, ya sea el valor de un sensor, un detector, ... y, actuando en consecuencia, escribir el valor deseado en las variables de salida de Home I/O. Para ello, primero se necesitará programar una función en Matlab para ambos casos. Una vez se tengan las funciones listas, éstas se ejecutarán desde Simulink mediante el uso de bloques “M-S-Function”. Un bloque ejecutará la función “read.m” y otro ejecutará la función “write.m”. En Simulink aparecerán los bloques de esta forma:

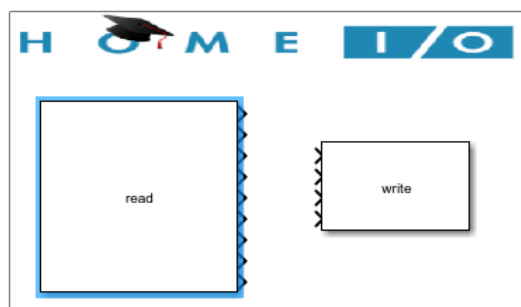


Figura 2-9. Bloques en Simulink que ejecutan “read.m” y “write.m”

2.2.3.1 Función de lectura de variables

En el Anexo D se encuentra el código de un ejemplo de función para la lectura de variables. La función de lectura consta fundamentalmente de dos partes:

- **function setup (block):** se utiliza para configurar el bloque. De esta función, lo que tiene mayor importancia es la definición del número de entradas y salidas que tendrá el bloque en Simulink y el período de muestreo, ya que son los parámetros modificables en el caso de que se quisieran leer más variables o cambiar el intervalo de tiempo entre dos muestras sucesivas. En el ejemplo de la siguiente figura se ha elegido que el bloque de Simulink tenga 6 salidas (ya que para lo que Home I/O es una entrada, para matlab es una salida) y un período de muestreo de 0.02 segundos.

```
function setup(block)

%% Register number of input and output ports
block.NumInputPorts = 0;
block.NumOutputPorts = 6;

%% Setup functional port properties to dynamically
block.SetPreCompPortInfoToDefaults;

block.OutputPort(1).Dimensions = 1;

%% Set block sample time to [0.1 0]
block.SampleTimes = [0.02 0];

%% Set the block simStateCompliance to default (i.e., same as a built-in block)
block.SimStateCompliance = 'DefaultSimState';
path = strcat(pwd, '\EngineIO.dll');
NET.addAssembly(path);

%% Register methods
block.RegBlockMethod('Outputs', @Output);
endfunction
```

Figura 2-10. Función setup de la función de lectura de variables de Home I/O

- **function output (block):** se utiliza para definir qué dispositivos se quieren leer y a qué puerto del bloque de Simulink corresponderá su valor. Se debe definir la dirección de la variable que se quiere leer, el tipo de dato que es (bit, float...) y el tipo de variable que es (si es una entrada, salida o una variable de memoria). En la siguiente figura hay un ejemplo en el que se lee la temperatura del cuarto de los niños (zona H), su luminosidad, el nivel de apertura de las persianas... Por ejemplo, en el caso de la temperatura, se puede observar que el dato es de tipo float, es una variable “input” y su dirección es la nº 58. Su valor se recogerá en el primer puerto del bloque de Simulink.

```
function Output(block)
EngineIO.MemoryMap.Instance.Update();

%Temperatura cuarto niños
block.OutputPort(1).Data = double(EngineIO.MemoryMap.Instance.GetFloat(58, EngineIO.MemoryType.Input).Value);
%Luminosidad cuarto niños
block.OutputPort(2).Data = double(EngineIO.MemoryMap.Instance.GetFloat(57, EngineIO.MemoryType.Input).Value);
%Nivel de apertura persianas cuarto niños
block.OutputPort(3).Data = double(EngineIO.MemoryMap.Instance.GetFloat(60, EngineIO.MemoryType.Input).Value);
%DETECTOR DE PUERTA 1 cuarto niños
block.OutputPort(4).Data = double(EngineIO.MemoryMap.Instance.GetBit(136, EngineIO.MemoryType.Input).Value);
%DETECTOR DE PUERTA 2 cuarto niños
block.OutputPort(5).Data = double(EngineIO.MemoryMap.Instance.GetBit(137, EngineIO.MemoryType.Input).Value);
%Potencia instantanea consumida
block.OutputPort(6).Data = double(EngineIO.MemoryMap.Instance.GetFloat(141, EngineIO.MemoryType.Memory).Value);

endfunction
```

Figura 2-11. Función output de la función de lectura de variables de Home I/O

Con esta estructura, se puede leer cualquier variable que se requiera del programa Home I/O.

2.2.3.2 Función de escritura de variables

En el Anexo E se encuentra el código de un ejemplo de función para la escritura de variables. Al igual que en el caso anterior, la función de escritura consta fundamentalmente de dos partes:

- function setup (block): se utiliza para configurar el bloque. De esta función, lo que tiene mayor relevancia es la definición del número de entradas y salidas que tendrá el bloque en Simulink y el período de muestreo, ya que son los parámetros modificables en el caso de que se quisieran leer más variables o cambiar el intervalo de tiempo entre dos muestras sucesivas. En el ejemplo de la siguiente figura se ha elegido que el bloque de Simulink tenga 4 entradas (ya que para lo que Home I/O es una salida, para matlab es una entrada) y un período de muestreo de 0.02 segundos. Es casi similar a la del apartado anterior.

```
function setup(block)

%% Register number of input and output ports
block.NumInputPorts = 4;
block.NumOutputPorts = 0;

%% Setup functional port properties to dynamically
block.SetPreCompPortInfoToDefaults;

block.InputPort(1).Dimensions = 1;

%% Set block sample time to [0.1 0]
block.SampleTimes = [0.02 0];

%% Set the block simStateCompliance to default (i.e., same as a built-in block)
block.SimStateCompliance = 'DefaultSimState';

path = strcat(pwd, '\EngineIO.dll');
NET.addAssembly(path);

%% Register methods
block.RegBlockMethod('Outputs', @Output);
endfunction
```

Figura 2-12. Función setup de la función de escritura de variables de Home I/O

- function output (block): se utiliza para definir sobre qué dispositivos queremos actuar (escribir un valor) y por qué puerto del bloque de Simulink Home I/O recibirá el valor. Se debe definir primero la dirección de la variable sobre la que se quiere escribir, el tipo de dato que es (bit, float...) y el tipo de variable que es (en este caso, siempre será una salida, porque son las únicas que permiten escritura). Entonces, el valor se dará a través de una entrada del bloque en Simulink. En la siguiente figura hay un ejemplo que permitiría controlar la calefacción, luz y persianas de la habitación de los niños de la vivienda (zona H).

```
function Output(block)
    %Calefactor cuarto niños
    Heater = EngineIO.MemoryMap.Instance.GetFloat(79, EngineIO.MemoryType.Output);
    Heater.Value = block.InputPort(1).Data;
    %Luz cuarto niños
    Luz = EngineIO.MemoryMap.Instance.GetFloat(78, EngineIO.MemoryType.Output);
    Luz.Value = block.InputPort(2).Data;
    %PersianasSubida cuarto niños
    PersianasUP = EngineIO.MemoryMap.Instance.GetBit(98, EngineIO.MemoryType.Output);
    PersianasUP.Value = block.InputPort(3).Data;
    %PersianasBajada cuarto niños
    PersianasDOWN = EngineIO.MemoryMap.Instance.GetBit(99, EngineIO.MemoryType.Output);
    PersianasDOWN.Value = block.InputPort(4).Data;

    EngineIO.MemoryMap.Instance.Update();
end
```

Figura 2-13. Función output de la función de escritura de variables de Home I/O

Con ambas funciones, se puede leer y actuar sobre cualquier variable de Home I/O.

2.2.3.4 Simulación en tiempo real

Al ser Home I/O un entorno de simulación en tiempo real, es necesario que Matlab sea capaz de simular en tiempo real también. Para ello, existe un paquete denominado “RealTime_Pacer”. Este paquete permite a Matlab simular en tiempo real en Simulink. Para poder utilizar el paquete, se debe añadir la carpeta “RealTime_Pacer” al path de Matlab. Una vez hecho esto, se tiene un nuevo bloque en Simulink disponible denominado “RealTime_Pacer”. Este bloque se incluye en el archivo de Simulink y a partir de ese momento, las simulaciones se realizarán en tiempo real.

El bloque de Simulink es el siguiente:

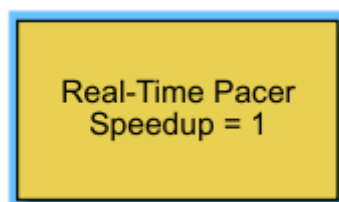


Figura 2-14. Bloque de Simulink RealTime_Pacer para la simulación en tiempo real

2.3 Microred simulada en Simulink

Como fuente de alimentación de energía eléctrica a la vivienda, se va a utilizar una microrred simulada en Simulink.

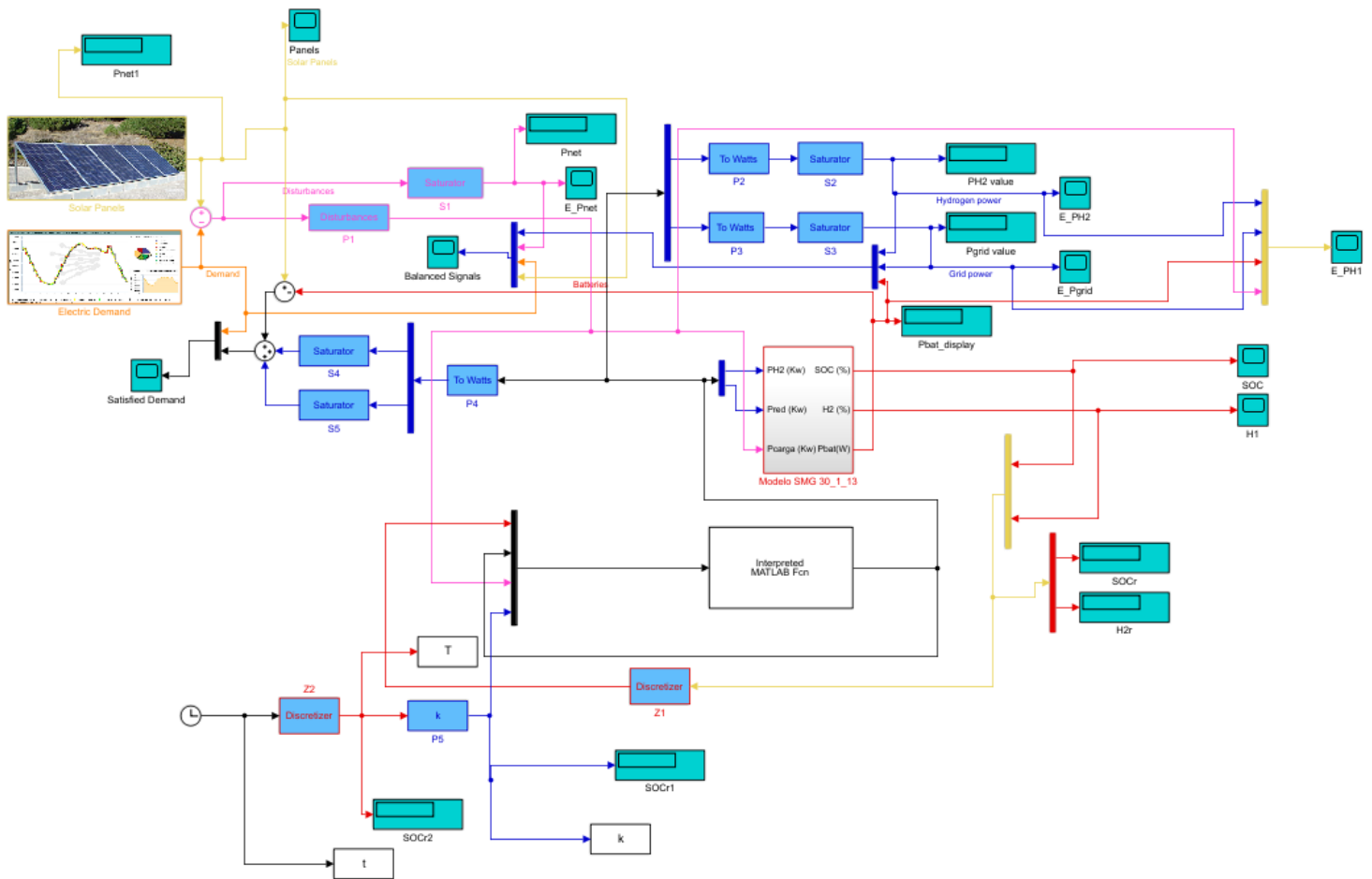


Figura 2-15. Microred simulada en Simulink

Esta es la microrred original que se estudia en el artículo [2]. Pero, a la que se realizará una serie de modificaciones, debido a que el tiempo de muestreo no es el adecuado para este trabajo y la forma de implementación del controlador MPC no es la óptima para las últimas versiones de Matlab.

2.3.1 Componentes de la microrred

La microrred consta de los siguientes componentes:

- Fuente de energía renovable: paneles solares, micro turbinas, ...
- Pila de combustible de membrana de intercambio protónico (pila de combustible PEM).
- Electrolizador de membrana polimérica protónica (electrolizador PEM).
- Tanque de hidruros metálicos, para la obtención de hidrógeno.
- Banco de baterías.

- Convertidores DC/DC.

2.3.2 Funcionamiento de la microred

La principal fuente de energía eléctrica vendrá dada por la energía renovable, que se encargará de alimentar a la carga, en este caso, la vivienda. Cualquier exceso de potencia puede ser dirigido al electrolizador o almacenado en el banco de baterías. Cuando la potencia obtenida gracias a la fuente de energía renovable no sea suficiente, tanto la pila de combustible como el banco de baterías pueden suministrar potencia eléctrica. Aunque se dispone de la posibilidad de trabajo de la microrred de forma aislada, ésta trabajará conectada a la red eléctrica pública en caso de que sea necesario eventualmente su uso.

El recorrido que sigue el hidrógeno en la microred es el siguiente:

1. Se dispone de un tanque de hidruros metálicos.
2. Del tanque de hidruros metálicos, se puede obtener hidrógeno mediante la electrólisis de membrana polimérica protónica, utilizando el electrolizador PEM. [3]
3. Obtenido el hidrógeno, se utiliza como combustible para conseguir energía eléctrica en la pila de combustible PEM. La energía eléctrica se obtiene mediante la oxidación electroquímica del hidrógeno y de la reducción electroquímica del oxígeno. [4]

Es importante destacar que el electrolizador y la pila de combustible no pueden trabajar simultáneamente.

2.3.3 Modelo lineal de la microred y restricciones

Como puede intuirse, el modelo real de la microrred está gobernado por subsistemas complejos y no lineales. Por otro lado, se necesita obtener un modelo lineal de la microrred para poder utilizar un controlador tipo MPC (Model Predictive Controller). Como consecuencia, la microrred ha sido linealizada en espacios de estados en tiempo discreto en torno a un punto de operación. [2]

El modelo consta de las siguientes características:

- Entradas: 2.
 - La potencia del electrolizador y la potencia de la pila de combustible ($P_{H_2}(k)$). Si $P_{H_2}(k)$ es mayor que cero, la pila de combustible es la que está trabajando generando potencia. Si es menor que cero, es el electrolizador el que está funcionando, absorbiendo potencia.
 - La potencia de la red eléctrica pública ($P_{grid}(k)$). Puede ser tanto mayor que cero, como menor que cero. Si es mayor que cero significará que se está haciendo uso de ella y por tanto, tendrá su coste económico. Si en cambio, es menor que cero, significará que se está exportando potencia a la red eléctrica.

Ambas potencias se miden en kW (kilovatios).
- Perturbaciones medibles: 1.
 - Tanto la potencia obtenida por la energía renovable ($P_{res}(k)$), como la potencia demandada por el consumidor ($P_{dem}(k)$) son variables sobre las que no se tiene control, pero si se pueden medir. La diferencia entre estas dos respectivamente, es la potencia neta ($P_{net}(k)$). Será considerada como perturbación a la microrred y se mide en kW (kilovatios).
- Estados o salidas: 2.
 - Estado de carga de la batería ($SOC(k)$). Se expresa en porcentaje (%).
 - Nivel de hidruros metálicos ($MHL(k)$). También se expresa en porcentaje (%).
- Tiempo de muestreo: 30 segundos.

- Punto de operación:

- $x^* = [50\%, 50\%]^T$

- $u^* = [0 \text{ kW}, 1.75 \text{ kW}]^T$

donde $x(k) = [SOC(k), MHL(k)]^T$ es el vector de estados o salidas; y $u(k) = [P_{H_2}(k), P_{grid}(k)]^T$ es el vector de entradas manipulables.

La planta tiene una variable adicional, la potencia de las baterías ($P_{batt}(k)$), que se controla indirectamente.

Una vez definidas sus características, el modelo discreto de la microred, para cada instante de tiempo, alrededor del punto de operación es el siguiente:

$$x(k+1) = x(k) + \begin{pmatrix} 8.1360 & 5.958 \\ -15.2886 & 0 \end{pmatrix} * u(k) + \begin{pmatrix} 5.958 \\ 0 \end{pmatrix} * w(k) \quad (1)$$

$$\text{donde } w(k) = P_{net}(k)$$

Pero, como a lo largo del trabajo se va a trabajar con modelos cuyo tiempo de muestreo es de 10 segundos, hay que transformar el modelo de la microred anterior. Para ello, se utiliza la función “d2d” de Matlab, permitiendo cambiar el tiempo de muestreo de un modelo lineal. Por tanto, se cambia el tiempo de muestreo del modelo de la microred de 30 segundos a 10 segundos.

Se obtiene entonces el siguiente nuevo modelo para la microred, que es con el que se trabajará a lo largo del trabajo:

$$x(k+1) = x(k) + \begin{pmatrix} 2.712 & 1.986 \\ -5.096 & 0 \end{pmatrix} * u(k) + \begin{pmatrix} 1.986 \\ 0 \end{pmatrix} * w(k) \quad (2)$$

La variables del sistema están sujetas a diferentes restricciones para evitar el daño de los equipos y garantizar su funcionamiento seguro. Por ello, es necesario limitar sus valores máximos y mínimos, asegurando así una mayor vida útil y un mejor funcionamiento de los equipos. Las restricciones son las siguientes:

$$-0.9 \text{ kW} \leq P_{H_2}(k) \leq 0.9 \text{ kW}, \quad (2a)$$

$$-2.5 \text{ kW} \leq P_{grid}(k) \leq 2 \text{ kW}, \quad (2b)$$

$$-20 \text{ W s}^{-1} \leq \Delta P_{H_2}(k) \leq 20 \text{ W s}^{-1}, \quad (2c)$$

$$-2.5 \text{ kW s}^{-1} \leq \Delta P_{grid}(k) \leq 2 \text{ kW s}^{-1}. \quad (2d)$$

A su vez, las variables de estado también tienen sus respectivas restricciones:

$$40 \% \leq SOC(k) \leq 90 \%, \quad (3a)$$

$$10 \% \leq MHL(k) \leq 90 \%. \quad (3b)$$

Finalmente, la potencia total que se le suministra a la carga, debe satisfacer el siguiente balance:

$$P_{\text{dem}}(k) = P_{H_2}(k) - P_{\text{batt}}(k) + P_{\text{grid}}(k) + P_{\text{res}}(k). \quad (4)$$

2.3.4 Controlador MPC de la microred

Conocido el modelo lineal de la microred, es inmediata la implementación de un controlador MPC para la gestión inteligente de la gestión de la potencia en los diferentes componentes de la microred. Para ello, se ha hecho uso del “Model Predictive Control toolbox” [5]. Es un toolbox de Matlab cuyo objetivo es el diseño y simulación de controladores usando tanto modelos lineales como modelos no lineales de control predictivo (MPC).

Para la creación del controlador, se ejecuta el código que aparece en la siguiente figura.

```
A=eye(2,2);
B=[2.712 1.986 1.986;-5.096 0 0];...
C=eye(2,2);
D=zeros(2,3);
modelo_MPC_microred=idss(A,B,C,D)
set(modelo_MPC_microred,'Ts',10)
modelo_MPC_microred=setmpcsignals(modelo_MPC_microred,'MD',[3],'MV',[1 2]);
MV=[struct('Min',-0.9,'Max',0.9,'RateMin',-0.02,'RateMax',0.02);...
     struct('Min',-2.5,'Max',2,'RateMin',-2.5,'RateMax',2)];...
MO=[struct('Min',40,'Max',90);struct('Min',10,'Max',90)];
MPC_microred=mpc(modelo_MPC_microred,10,10,2,[],MV,MO,[])
```

Figura 2-16. Código para la creación del controlador MPC de la microred

El código implementa el modelo lineal de la ecuación (2) y tiene en cuenta las diferentes restricciones para cada variable que se han detallado en las ecuaciones (2a), (2b), (2c), (2d), (3c) y (3d).

En cuanto a las características del controlador, son las siguientes:

- Tiempo de muestreo: 10 segundos
- Horizonte de predicción: 10
- Horizonte de control: 2
- Pesos:
 - Para las entradas manipulables:
 - Para $P_{H_2}(k)$: 0
 - Para $P_{\text{grid}}(k)$: 0
 - Para el incremento de las entradas manipulables:
 - Para $P_{H_2}(k)$: 0.1
 - Para $P_{\text{grid}}(k)$: 0.1

- Para las variables de salida:
 - Para $SOC(k)$: 1
 - Para $MHL(k)$: 1

Para implementar el controlador en Simulink, hay directamente un bloque denominado “MPC Controller” que puede utilizar un controlador ya definido en el Workspace, como es el caso. Las referencias que se utilizan para el control en bucle cerrado de la microred son las siguientes:

- $SOC_{ref} = 65\%$
- $MHL_{ref} = 40\%$

Por tanto, el diagrama de bloques en Simulink que simula la microred con el que se va a trabajar finalmente es el siguiente:

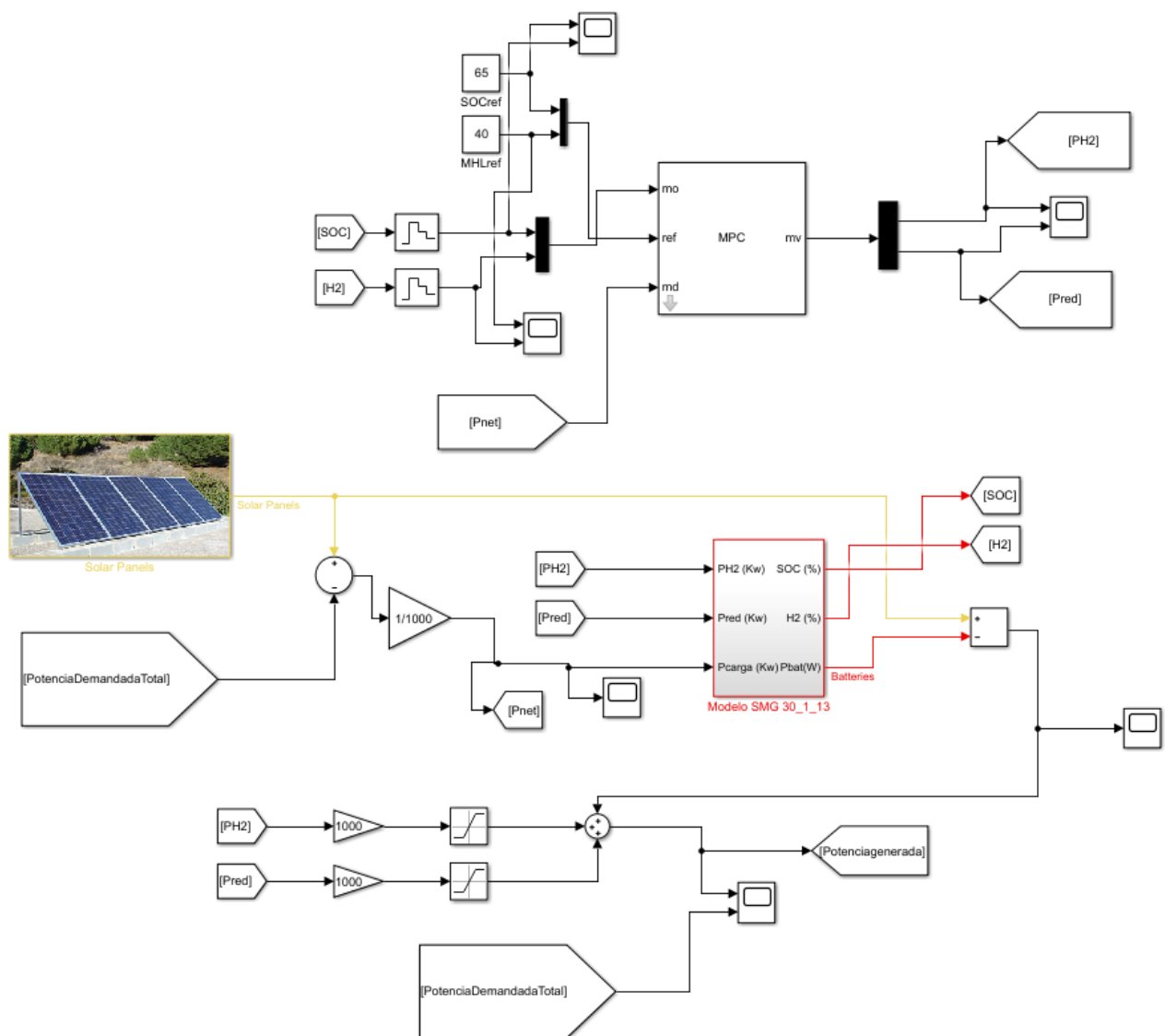


Figura 2-17. Microred definitiva simulada en Simulink con su respectivo controlador MPC

En este trabajo, la carga va a ser la potencia demandada total que realice la casa en cada instante de tiempo.

3 MODELOS Y CONTROLADORES

El objetivo del control de Home I/O mediante el uso de Matlab Simulink para este trabajo es la gestión inteligente de la energía eléctrica en una casa automatizada cuya energía eléctrica viene dada por microred o en la que se quiere limitar la potencia por algún motivo. Para ello, se modificará de forma automática la potencia de aparatos eléctricos de la vivienda, pudiéndose llegar a apagar.

La primera parte del trabajo se va a centrar en la habitación de los niños (zona H de la vivienda), al que al partir de ahora se va a denominar **Sistema A**. En esta habitación, se encuentra un calefactor de 1000 W, una luz regulable de 40 W y unas persianas cuya acción consume 10 W. A su vez, se encuentra un termostato que permite conocer la temperatura de la habitación en todo momento, y un sensor de luminosidad que mide la luminosidad de la habitación. El control se centrará en el calefactor ya que es el aparato eléctrico que tiene mayor impacto en el consumo. Se comparará la utilización de un Controlador Predictivo Basado en Modelo (MPC), uniendo el modelo de la microred con la del calefactor, con la utilización de otro controlador MPC, pero en este caso, sin incluir el modelo del calefactor, y en el que el control de la temperatura se realiza mediante un controlador todo/nada con histéresis.

La segunda parte del trabajo se va a centrar en el control basado en reglas aplicado a todos los calefactores de la vivienda (disponibles en la Tabla 2-2), al que se va a denominar **Sistema B**. El control de los calefactores también se realizará mediante controladores todo/nada con histéresis. En este sistema no se va a realizar uso de la microred y se va a suponer que solo se dispone de la red eléctrica pública para su alimentación. Se impondrá una potencia máxima a elección del usuario, que se podría modificar cuando él lo requiriese.



Figura 3-1. Habitación de los niños (zona H)

3.1 Control predictivo basado en modelo (calefactor + microred) para el Sistema A

Previamente a la implementación del controlador, se ha realizado una identificación del sistema a controlar; en este caso, la calefacción. Para ello, primero hay que comprender el funcionamiento de la transmisión de calor en Home I/O, las distintas variables que afectan a la temperatura de la habitación y realizar una hipótesis de trabajo para nuestro modelo.

3.1.1 Diseño del modelo

Las distintas variables meteorológicas que afectan a la temperatura de la habitación en Home I/O son las siguientes:

- Temperatura del ambiente
- Dirección del viento
- Velocidad del viento
- Nubosidad
- Humedad

Se va a definir entonces unas condiciones de operación para estas variables en el modelado, siendo las siguientes:

- Temperatura ambiental: variable entre 10 y 20 °C
- Velocidad del viento: 5 km/h. Es la mínima posible que permite establecer el programa, ya que nos interesaría que fuera nula para el modelo.
- Dirección del viento: al tener el viento una velocidad tan baja, esta variable afectará de forma inapreciable al modelo.
- Nubosidad: 1%. También es el valor mínimo que permite establecer el programa.
- Humedad: 40%.

Se ha realizado un modelo de caja negra de espacio de estados donde las variables de entrada son:

- Grado de activación del calefactor (0-10V) ($V_C(k)$)
- Temperatura exterior (en Kelvin) ($T_{Ext}(k)$)
- Las temperaturas de las zonas de la vivienda colindantes, estas son, las zonas G e I (en Kelvin) ($T_G(k)$ y $T_I(k)$ *respectivamente*)

Las temperaturas de las zonas G e I y la temperatura exterior son entradas exógenas o perturbaciones, cuyo valor no se va a manipular. En cambio, el grado de activación del calefactor será la única entrada manipulable. La variable de salida es la temperatura de la zona H.

El modelo es de orden 1 y se han realizado varios experimentos para su identificación. Se han medido las diferentes entradas y la salida durante 1 día en las condiciones de operación aplicando diferentes saltos al calefactor con un tiempo de muestreo de 10 segundos, obteniéndose unos EMC de 0.02842, 0.02769 y 0.03411. El punto de operación para el grado de activación del calefactor es de 0V y de 288 K para la temperatura de la zona H.

El modelo se ha obtenido utilizando el “System Identification Toolbox” de Matlab. [6]

3.1.2 Validación del modelo

Con el modelo ya definido, es necesario validarlo. Para ello, se realizan tres experimentos más, de forma que los nuevos datos sean diferentes de los que se han utilizado para identificar el modelo, para demostrar su validez.

- 1º experimento. Los datos obtenidos fueron los siguientes:

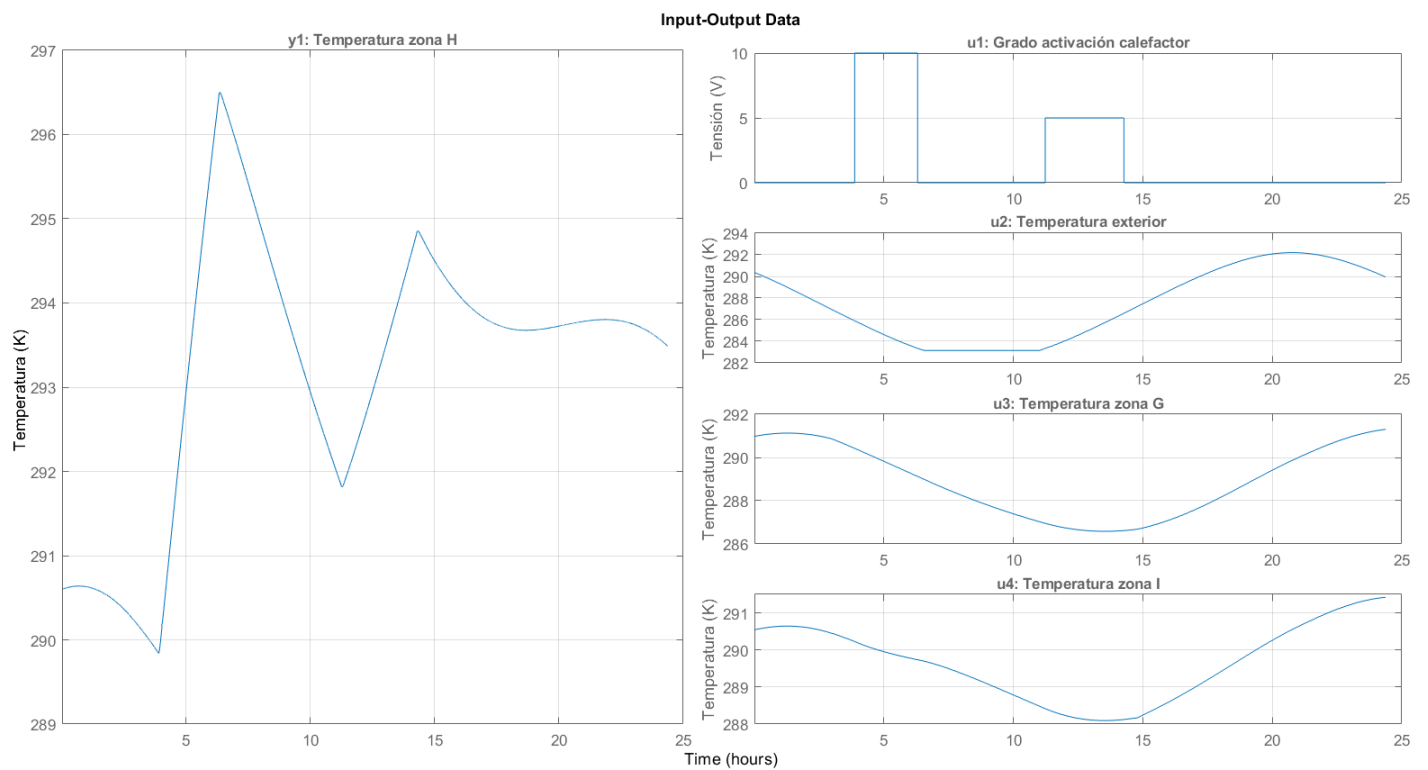


Figura 3-2. Datos obtenidos del experimento 1 para la validación del modelo

Comparandolo con el modelo:

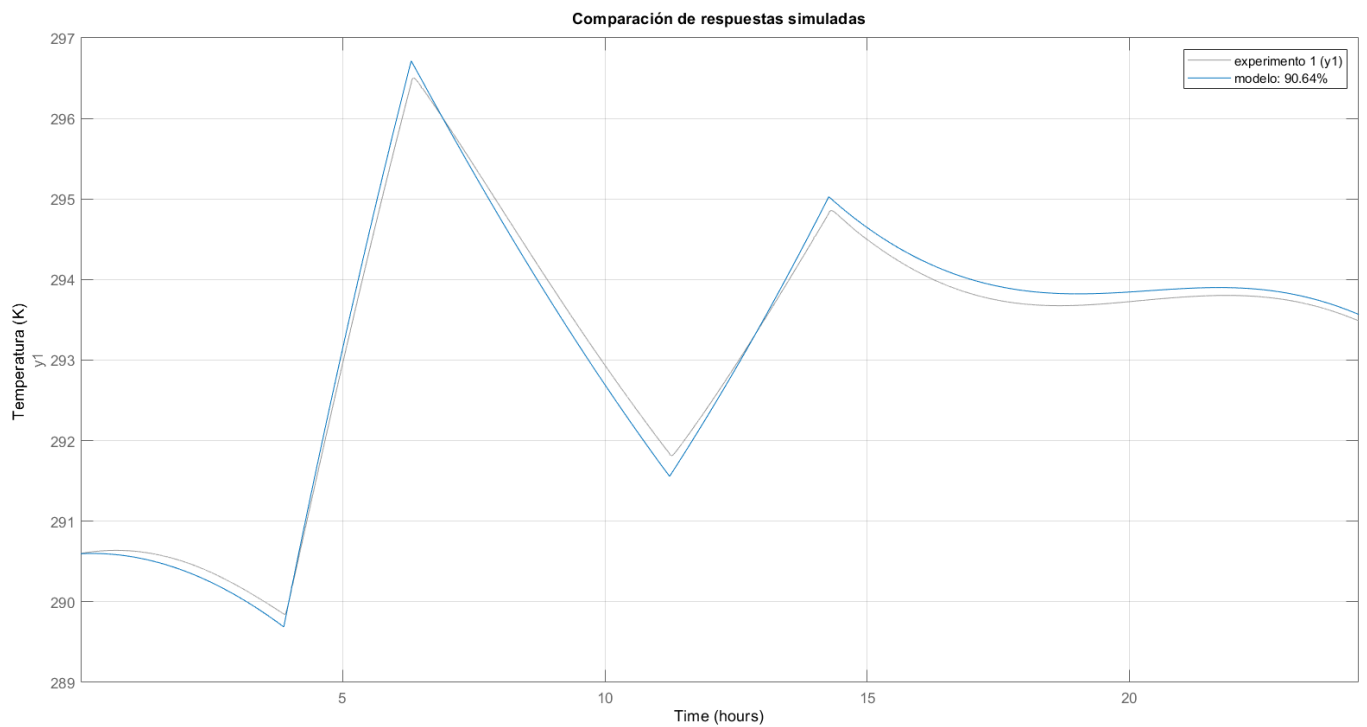


Figura 3-3. Comparación de respuestas simuladas del experimento 1 y el modelo

- 2º experimento. Los datos obtenidos fueron los siguientes:

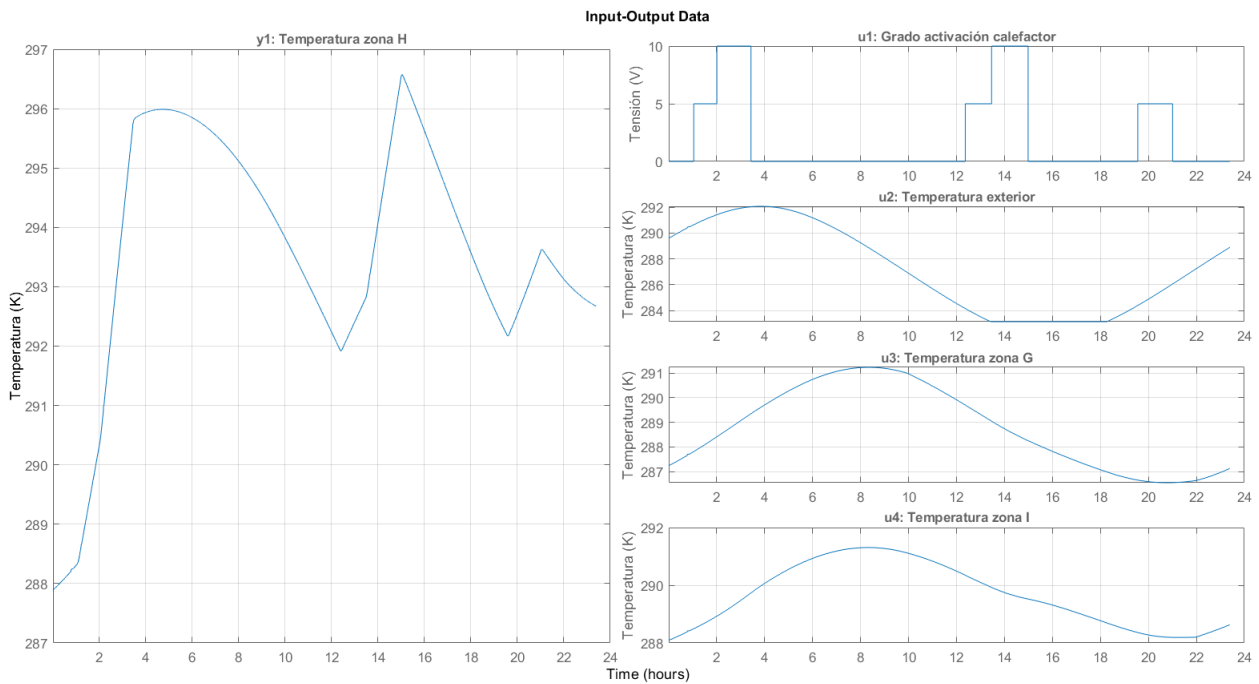


Figura 3-4. Datos obtenidos del experimento 2 para la validación del modelo

Comparandolo con el modelo:

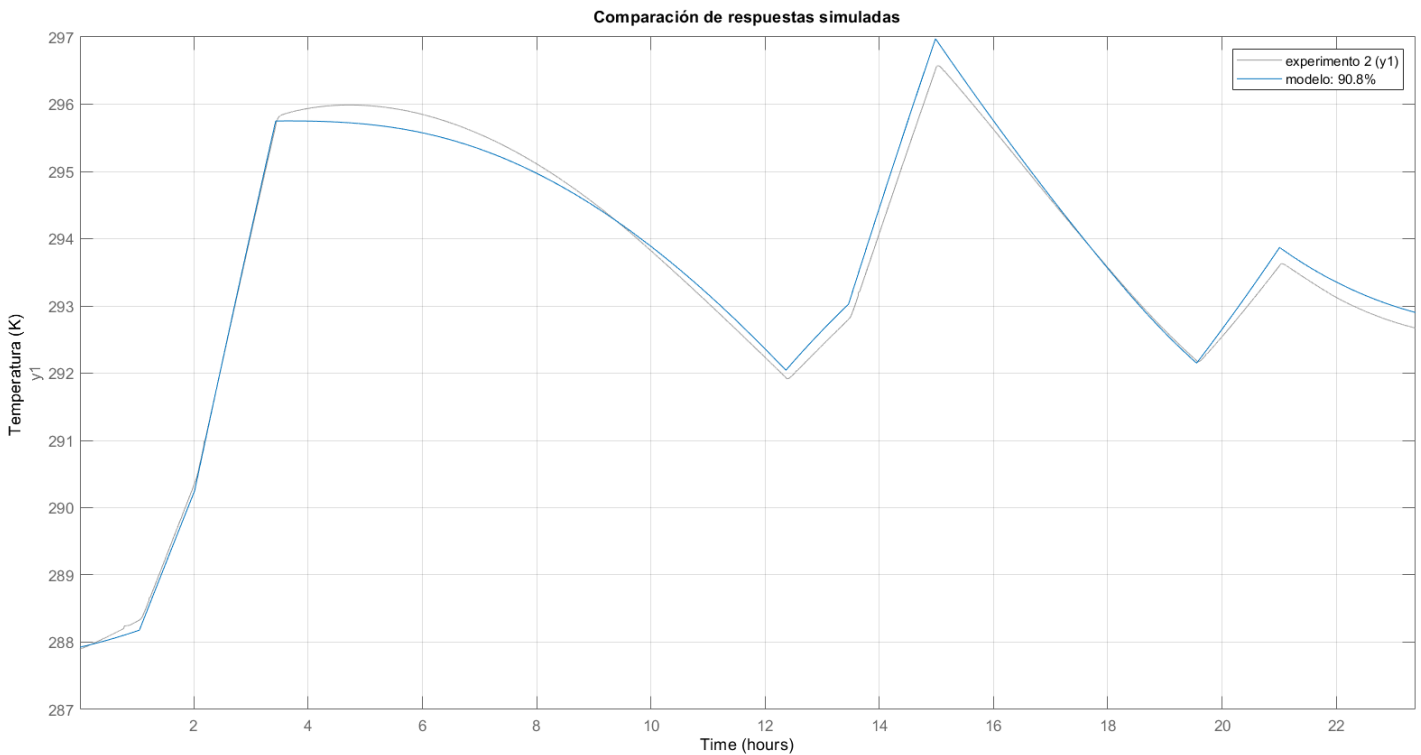


Figura 3-5. Comparación de respuestas simuladas del experimento 2 y el modelo

- 3º experimento. Los datos obtenidos fueron los siguientes:

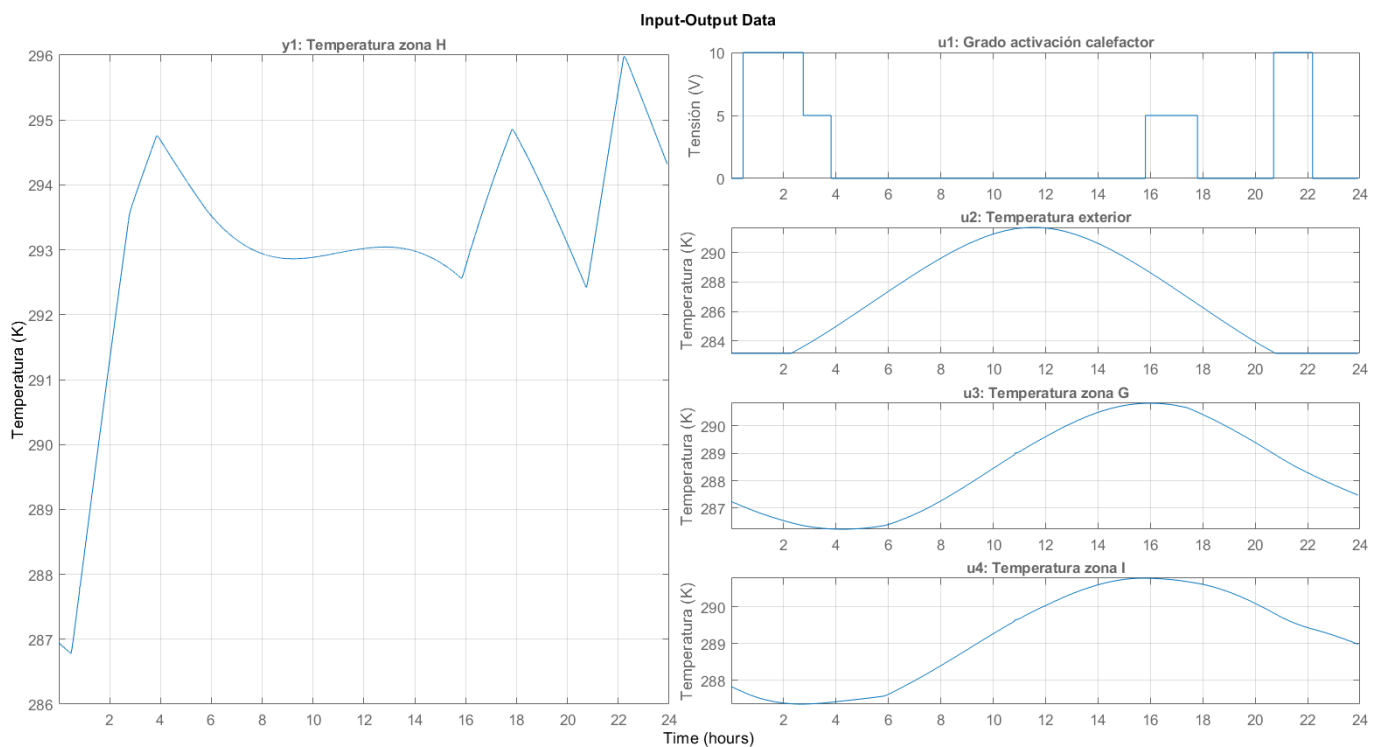


Figura 3-6. Datos obtenidos del experimento 3 para la validación del modelo

Comparandolo con el modelo:

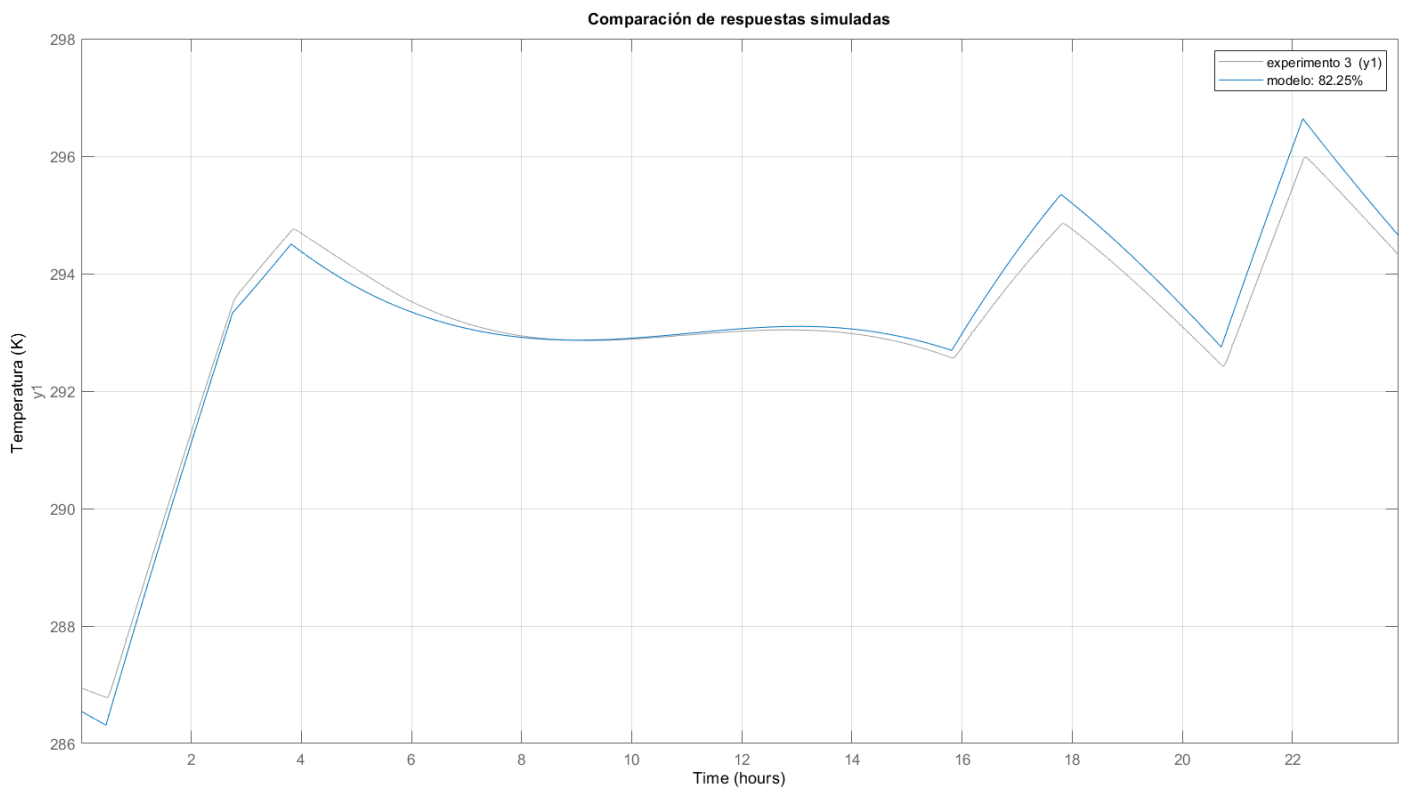


Figura 3-7. Comparación de respuestas simuladas del experimento 3 y el modelo

Como se puede observar en las figuras anteriores, el modelo tiene una precisión muy alta (80-90%) y se puede considerar que el modelo es válido.

3.1.3 Unión del modelo lineal de la microred con el modelo del calefactor

Una vez se ha obtenido el modelo lineal que relaciona la temperatura de la habitación de los niños con el grado de activación del calefactor, se va a incorporar al modelo lineal de la microred el modelo del calefactor. De este modo, se tienen ambos modelos en uno. Para ello, primero es necesario que ambos modelos tengan el mismo tiempo de muestreo.

Al modelo de la microred del apartado 2.3.3 Modelo lineal de la microred y restricciones la ecuación (2), le incluimos el modelo del calefactor obteniendo lo siguiente:

$$x(k+1) = x(k) + \begin{pmatrix} 2.712 & 1.986 & -0.1986 \\ -5.096 & 0 & 0 \\ 0 & 0 & 0.001307 \end{pmatrix} * u(k) + \begin{pmatrix} 1.986 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.0003643 & 0.0001531 & -0.0003776 \end{pmatrix} * w(k)$$

- Donde $x(k) = [SOC(k), MHL(k), T_H(k)]^T$
- Donde $u(k) = [P_{H_2}(k), P_{grid}(k), V_C(k)]^T$
- Donde $w(k) = [P_{net}(k), T_{Ext}(k), T_G(k), T_I(k)]^T$

El coeficiente (1,3) de la matriz que multiplica $u(k)$ se ha obtenido siguiendo el siguiente razonamiento. Observando el modelo lineal de la microred, se deduce que, para obtener unidades de potencia en el primer estado, se multiplica por el coeficiente 5.958 (para un tiempo de muestreo de 30 segundos). Ahora, el coeficiente ha pasado a ser 1.986 (para un tiempo de muestreo de 10 segundos). La relación entre la tensión de entrada del calefactor y la potencia que consume es lineal. En concreto, se obtiene multiplicando por 100 la tensión de entrada, y se obtiene la potencia que consume el calefactor en W. Como la unidad de potencia en el primer estado es en kW, hay que dividir el valor resultante entre 1000. Entonces, se obtiene que la relación entre el estado de la carga de la batería y la tensión de entrada del calefactor es la siguiente:

$$1.986 * \frac{100}{1000} = 0.1986$$

Es como si se tuviera en realidad lo siguiente:

$$\dots \begin{pmatrix} 2.712 & 1.986 & -1.986 \\ -5.096 & 0 & 0 \\ 0 & 0 & 0.001307 \end{pmatrix} * u(k) + \dots \text{donde } u(k) \text{ fuese } = [P_{H_2}(k), P_{grid}(k), P_C(k)]^T$$

Se le coloca un signo negativo, ya que la potencia del calefactor siempre va a hacer que disminuya la carga de la batería.

El resto de los coeficientes nuevos: 0.001307, 0.0003643, 0.0001531... se obtienen del modelo del calefactor.

3.1.4 Implementación del controlador y del modelo en Simulink

Para la implementación del controlador, se sigue un procedimiento similar al que se realizó para la microred en la **¡Error! No se encuentra el origen de la referencia..** En este caso, el script de Matlab tiene el siguiente código:

```
A=eye(3,3);
B=[2.712 1.986 -0.1986 1.986 0 0 0;-5.096 0 0 0 0 0;0 0 0.001037 ...
    0 0.0003643 0.0001531 -0.0003776];...

C=eye(3,3);
D=zeros(3,7);
modelo_MPC=idss(A,B,C,D)
set(modelo_MPC,'Ts',10)
modelo_MPC=setmpcsignals(modelo_MPC,'MD',[4 5 6 7],'MV',[1 2 3]);
MV=[struct('Min',-0.9,'Max',0.9,'RateMin',-0.02,'RateMax',0.02);...
    struct('Min',-2.5,'Max',2,'RateMin',-2.5,'RateMax',2);...
    struct('Min',0,'Max',10,'RateMin',-inf,'RateMax',inf)];
MO=[struct('Min',40,'Max',90);struct('Min',10,'Max',90);...
    struct('Min',273,'Max',298)];
Weights=struct('ManipulatedVariables',[0 0 0],'OutputVariables',[1 1 1]);
controlador_MPC=mpc(modelo_MPC,10,10,2,Weights,MV,MO,[])
```

Figura 3-8. Código para la creación del controlador MPC del modelo de la microred + calefactor

Las restricciones para las variables de la microred se mantienen similares. Pero, se introducen nuevas restricciones para las variables del modelo del calefactor:

$$0 \text{ V} \leq V_C(k) \leq 10 \text{ V}$$

$$273 \text{ K} \leq T_H(k) \leq 298 \text{ K}$$

Al incremento del grado de activación del calefactor, no se impone ninguna restricción.

En cuanto a las características del controlador, son las siguientes:

- Tiempo de muestreo: 10 segundos
- Horizonte de predicción: 10
- Horizonte de control: 2
- Pesos:
 - Para las entradas manipulables:
 - Para $P_{H_2}(k)$: 0
 - Para $P_{grid}(k)$: 0
 - Para $V_C(k)$: 0
 - Para el incremento de las entradas manipulables:

- Para $P_{H_2}(k)$: 0.1
- Para $P_{grid}(k)$: 0.1
- Para $V_C(k)$: 0.1
- Para las variables de salida:
 - Para $SOC(k)$: 1
 - Para $MHL(k)$: 1
 - Para $T_H(k)$: 1

La implementación del controlador en Simulink también es similar:

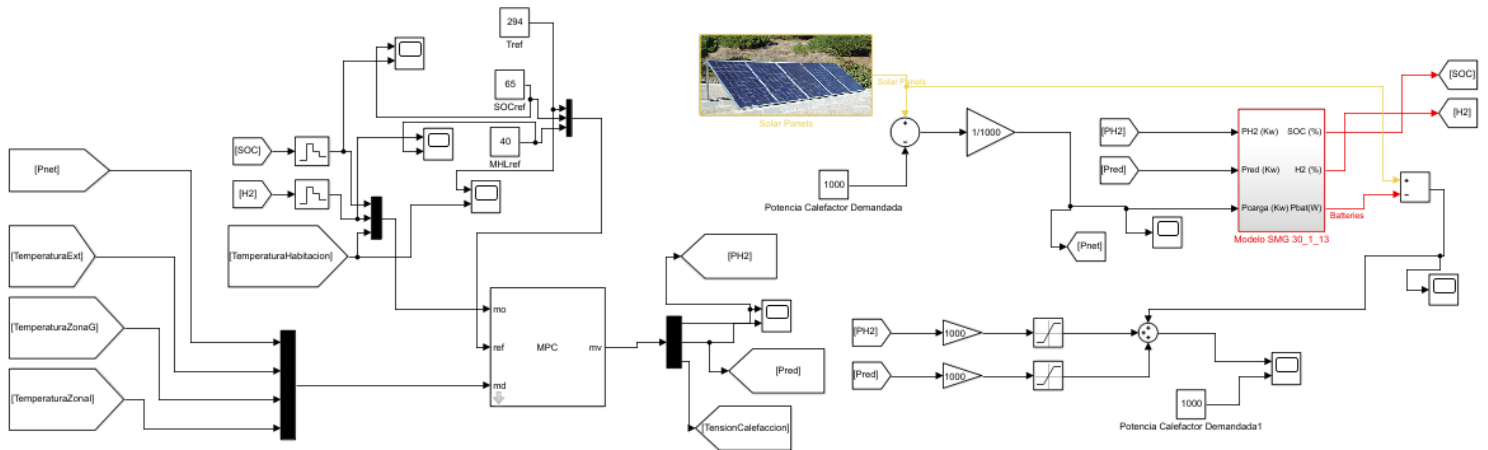


Figura 3-9. Modelo en Simulink del controlador MPC de la microred+calefactor

Entonces, el diagrama de bloques en Simulink completo (con la lectura y escritura de variables en Home I/O) es el siguiente:

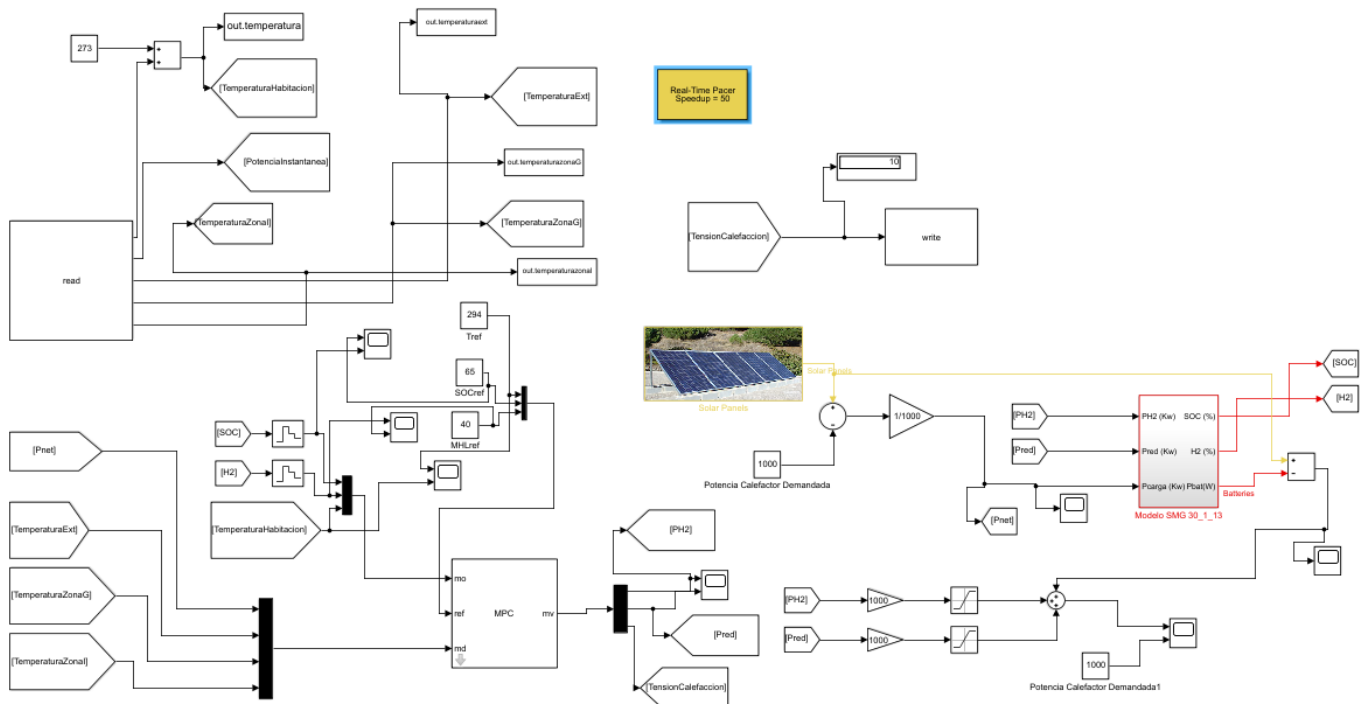


Figura 3-10. Modelo en Simulink del Sistema A con el controlador MPC completo (modelo microred+calefactor)

3.2 Control basado en reglas para el Sistema B

El objetivo, mencionado anteriormente, es el de controlar todos los calefactores disponibles de la vivienda de forma que, si la potencia total instantánea consumida por la vivienda supere la máxima impuesta por el usuario, se empiecen a reducir la potencia o se apaguen algunos calefactores de la vivienda en función de lo que sea necesario según un orden de prioridad, establecido previamente por el usuario.

3.2.1 Lectura de las temperaturas de las habitaciones

El primer paso a realizar es la lectura de las temperaturas de los diferentes termostatos disponibles en cada habitación equipada con calefactor y de la potencia instantánea total consumida por la vivienda. Para ello, consultando el anexo A y C, se pueden obtener las diferentes direcciones. Entonces, se modifica adecuadamente la función “read.m”. La función está disponible completamente en el anexo F para más información. En Simulink, se tendría lo siguiente:

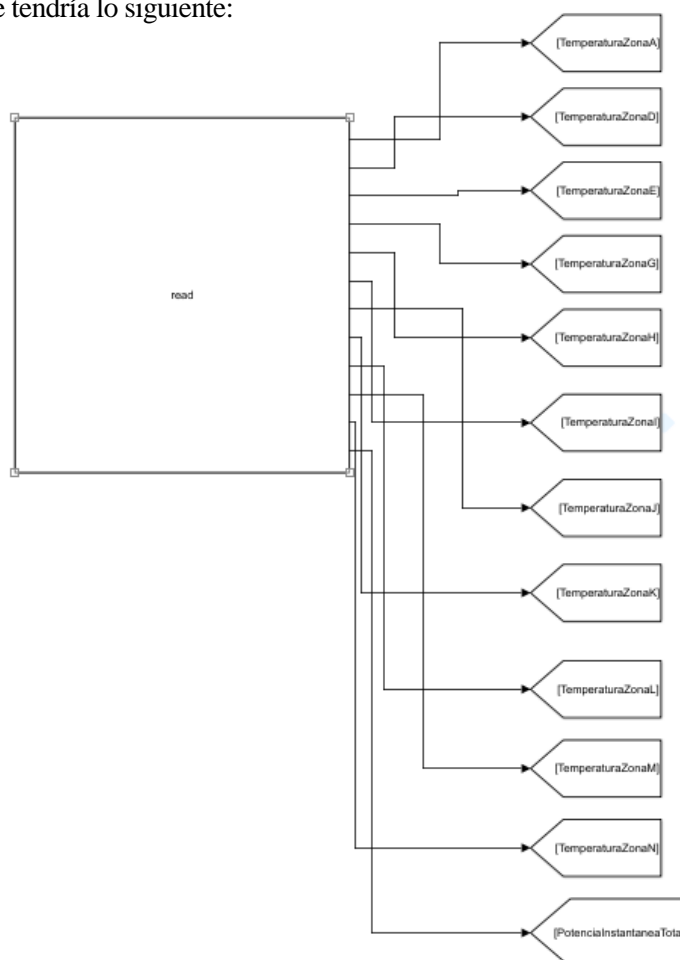


Figura 3-11. Bloque de lectura de las temperaturas de todas las habitaciones en Simulink

A cada salida del bloque le asignamos una señal con la correspondiente etiqueta para una mejor comprensión y organización de las diferentes variables.

3.2.2 Asignación de referencias de temperatura para cada habitación

Se asigna a cada habitación una referencia de temperatura:

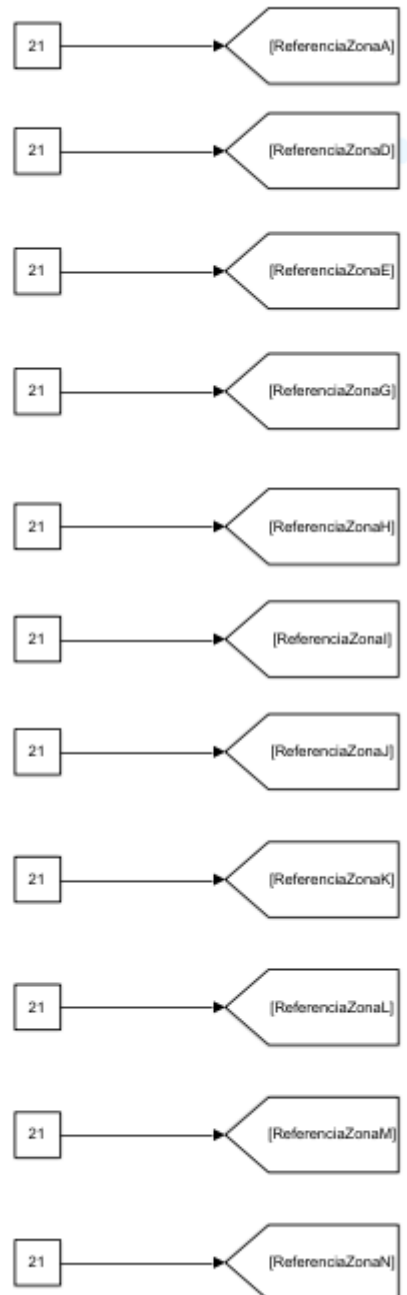


Figura 3-12. Asignación de referencia de temperatura para cada habitación en Simulink

3.2.3 Controladores on/off con histéresis

3.2.3.1 Esquema y funcionamiento del controlador

Para cada habitación con calefactor, el esquema que tiene el controlador en Simulink se puede observar en la siguiente figura:

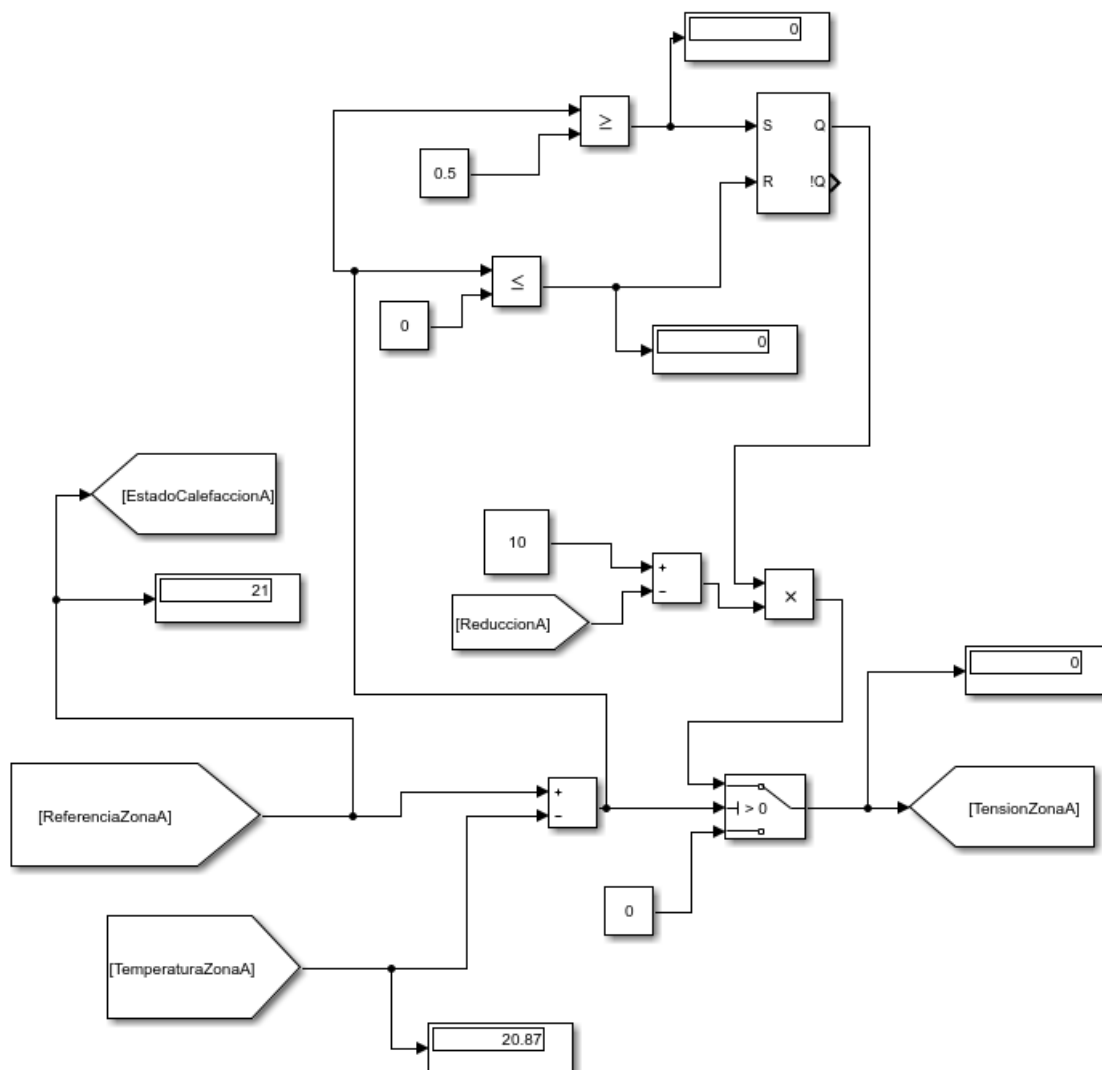


Figura 3-13. Controlador on/off con histéresis para controlar la temperatura de cada habitación en Simulink

En un principio, el controlador funciona de la siguiente forma: Si la diferencia entre la temperatura de referencia y la temperatura real de la correspondiente habitación es mayor que 0, la calefacción se enciende y funciona a la máxima tensión posible (10 V). Si es menor o igual que 0, la calefacción se apaga.

Esta tensión de trabajo del calefactor podría ser modificada por voluntad del usuario dependiendo de sus necesidades, pero en este caso, se ha definido que todos los calefactores trabajen a su tensión máxima (10 V) cuando estén funcionando.

Pero esta forma de controlar la temperatura acarrea una cantidad de conmutaciones muy alta que reduce la vida útil del conmutador. Por ello, se aplica una banda de histéresis. El controlador, una vez llegue a la temperatura de referencia (21 °C) y se apague, no se volverá a encender hasta que se alcance la temperatura de 20,5 °C o menos. Esto provoca una disminución considerable de conmutaciones que alargan la vida útil del aparato eléctrico.

Para programar esta característica, en Simulink, se ha hecho uso de los biestables RS. La tabla de transición de

estados según la temperatura para este caso es la siguiente:

Temperatura	S	R	Q	Tensión entrada calefactor (V)
$T \leq 20,5 \text{ }^{\circ}\text{C}$	1	0	1	10
$20,5^{\circ}\text{C} < T < 21 \text{ }^{\circ}\text{C}$	0	0	1	10
$T \geq 21 \text{ }^{\circ}\text{C}$	0	1	0	0
$21^{\circ}\text{C} > T > 20,5 \text{ }^{\circ}\text{C}$	0	0	0	0
$T \leq 20,5 \text{ }^{\circ}\text{C}$	1	0	1	10

Tabla 3-1. Tabla de transición de estados de los controladores on/off con histeresis

Esta transición se repetirá cíclicamente una vez llegue a la referencia hasta que se apagase el calefactor. La tensión final de entrada al calefactor (señal “TensiónZonaA”) será $10 \cdot Q$.

La señal “EstadoCalefaccionA” indicará si la calefacción está apagada o encendida. Si la referencia es 0, la señal “EstadoCalefaccionA” valdrá 0 y se entenderá que está apagada. Cuando la referencia tenga un valor distinto de 0 y mayor que la temperatura de la habitación en ese momento, estará encendida. En cuanto a la otra señal “ReduccionA”, se explicará más adelante.

3.2.4 Modulación automática de los calefactores

Para apagar o modificar la tensión con la que trabajan los diferentes calefactores, primero hay que conocer si están apagados o encendidos. Para ello, existen las señales mencionadas anteriormente: “EstadoCalefaccionA”. Hay una señal para cada calefactor que indicará su estado.

Para la implementación del programa se ha diseñado una nueva función denominada “modulación_y_apagado.m”, cuyo código viene completo en el anexo G pero que se explicará a continuación parte por parte:

- Function setup (block): en esta función se especifica el número de entradas y salidas que tiene el bloque de Simulink, y sus dimensiones. También se establece el período de muestreo. El resto de propiedades se dejan definidas por defecto.

En este caso, se necesitan 12 entradas y 11 salidas, y se utilizará un período de muestreo de 0.1 segundos. Los puertos se configuran por defecto, y el método utilizado es el de “Outputs”.

```
function setup(block)

%% Register number of input and output ports
block.NumInputPorts = 12;
block.NumOutputPorts = 11;

%% Setup functional port properties to dynamically
block.SetPreCompPortInfoToDefaults;

block.InputPort(1).Dimensions = 1;
block.OutputPort(1).Dimensions = 1;

%% Set block sample time to [0.1 0]
block.SampleTimes = [0.1 0];

%% Set the block simStateCompliance to default (i.e., same as a built-in block)
block.SimStateCompliance = 'DefaultSimState';

%% Register methods
block.RegBlockMethod('Outputs', @Output);
endfunction
```

Figura 3-14. Funcion setup de la función “modulado_y_apagado.m”

- Function output (block): se definen las entradas que tiene el bloque. Se pueden apreciar en la siguiente figura:

```
function Output(block)
%% lectura entradas
Diferencia=block.InputPort(1).Data;
EstadoA=double(block.InputPort(2).Data);
EstadoD=block.InputPort(3).Data;
EstadoE=block.InputPort(4).Data;
EstadoG=block.InputPort(5).Data;
EstadoH=block.InputPort(6).Data;
EstadoI=block.InputPort(7).Data;
EstadoJ=block.InputPort(8).Data;
EstadoK=block.InputPort(9).Data;
EstadoL=block.InputPort(10).Data;
EstadoM=block.InputPort(11).Data;
EstadoN=block.InputPort(12).Data;
%%
```

Figura 3-15. Definición de entradas en la función output de la function “modulado_y_apagado.m”

Las entradas son los diferentes estados de cada calefactor (si está apagado o encendido), ya que será necesario conocer aquellos que estén encendidos para actuar sobre ellos, y la diferencia entre la potencia instantánea total consumida por la casa y la potencia máxima que se ha puesto como límite.

Después, se declaran una serie de variables globales denominadas “ReduccionX” donde X es la letra de la zona de la vivienda a la que corresponden. En el controlador del calefactor, a la hora de introducir la tensión de entrada de cada calefactor, se le restaba a 10 la variable “ReduccionX”. Estas variables controlarán la potencia con la que funcione el calefactor, ya que se cree que es preferible reducir la potencia de 20% en 20% que apagar el calefactor directamente. Las variables son globales para poder darle un valor inicial de 0 a cada una de ellas en un script de Matlab de inicialización antes de empezar la simulación.

El objetivo del algoritmo es comprobar que si la variable “Diferencia” es mayor que 0, empezar a reducir la potencia de los calefactores ordenados según un orden de prioridad establecido por el usuario según sus necesidades. En este caso, se ha seguido el siguiente orden de menor a mayor prioridad:

- La lavandería (zona M). Es una zona de la casa donde se permanece poco tiempo y es poco relevante.
- La sala de estar (zona A). Es una zona donde se reúnen los convivientes de la casa, por lo que es poco probable que se utilicen más calefactores simultáneamente, llegando a alcanzar la potencia máxima. Y si se da el caso, será porque entonces la sala de estar estará vacía o en la que solo se encontrará una persona, que sabiendo la potencia que consume, es conveniente que se considere de poca prioridad.
- Hall de entrada y pasillo de los dormitorios (zonas E, G respectivamente). Son zonas de la vivienda de tránsito, en las que no se suele permanecer mucho tiempo.
- Cocina (zona D). Tiene cierta prioridad, pero sigue siendo inferior al de las siguientes zonas.
- Oficina (zona N). Es un lugar de trabajo donde es probable que solo permanezca una persona, y el calefactor de la oficina tiene un consumo alto. Se podría considerar que tiene una prioridad media.
- Los dormitorios (zonas H, J, L). Son las zonas de la vivienda en las que más tiempo se permanece y también zonas de descanso, donde se debe asegurar la comodidad del usuario.
- Por último, se encuentran los baños (zonas I, K). Tienen máxima prioridad, ya que tienen poco consumo y no se utilizan durante un período largo de tiempo. El uso del calefactor en esta estancia es esencial para el confort del usuario, sobre todo en invierno.

Las diferentes zonas de la vivienda y la potencia que consume cada calefactor están disponibles en la Tabla 2-2. En la siguiente figura se observa la implementación de este algoritmo siguiendo el orden de prioridad recién mencionado y la reducción de potencia de 20% en 20 %.

```

%%
global ReduccionA
global ReduccionD
global ReduccionE
global ReduccionG
global ReduccionH
global ReduccionI
global ReduccionJ
global ReduccionK
global ReduccionL
global ReduccionM
global ReduccionN
if (Diferencia>0)
    if (ReduccionM<10 && EstadoM>0)
        ReduccionM=ReduccionM+2;
    elseif (ReduccionA<10 && EstadoA>0)
        ReduccionA=ReduccionA+2;
    elseif (ReduccionE<10 && EstadoE>0)
        ReduccionE=ReduccionE+2;
    elseif (ReduccionG<10 && EstadoG>0)
        ReduccionG=ReduccionG+2;
    elseif (ReduccionD<10 && EstadoD>0)
        ReduccionD=ReduccionD+2;
    elseif (ReduccionN<10 && EstadoN>0)
        ReduccionN=ReduccionN+2;
    elseif (ReduccionH<10 && EstadoH>0)
        ReduccionH=ReduccionH+2;
    elseif (ReduccionJ<10 && EstadoJ>0)
        ReduccionJ=ReduccionJ+2;
    elseif (ReduccionL<10 && EstadoL>0)
        ReduccionL=ReduccionL+2;
    elseif (ReduccionI<10 && EstadoI>0)
        ReduccionI=ReduccionI+2;
    elseif (ReduccionK<10 && EstadoK>0)
        ReduccionK=ReduccionK+2;
    end
end

```

Figura 3-16. Algoritmo secuencial de apagado en la función output de la function “modulado_y_apagado.m”

El valor de entrada de tension de los calefactores es de 0 a 10 V. Como consecuencia, las variables “Reduccion” podrán tomar valores entre 0 y 10 también. La relación entre la tension de entrada del calefactor y la potencia que consume es lineal. Por lo que el hecho de aumentar las variables “Reduccion” de 2 en 2 es equivalente a reducir la potencia de 20% en 20%.

Hay que tener en cuenta el caso en el que la potencia máxima cambie, y se tengan que volver a encender calefactores que habían visto reducidos su potencia o, directamente apagados. Para ello, se incluye el código de la siguiente figura:

```

elseif (Diferencia<-300)
    if (ReduccionK>0 && EstadoK>0)
        ReduccionK=ReduccionK-2;
    elseif (ReduccionI>0 && EstadoI>0)
        ReduccionI=ReduccionI-2;
    elseif (ReduccionL>0 && EstadoL>0)
        ReduccionL=ReduccionL-2;
    elseif (ReduccionJ>0 && EstadoJ>0)
        ReduccionJ=ReduccionJ-2;
    elseif (ReduccionH>0 && EstadoH>0)
        ReduccionH=ReduccionH-2;
    elseif (ReduccionN>0 && EstadoN>0)
        ReduccionN=ReduccionN-2;
    elseif (ReduccionD>0 && EstadoD>0)
        ReduccionD=ReduccionD-2;
    elseif (ReduccionG>0 && EstadoG>0)
        ReduccionG=ReduccionG-2;
    elseif (ReduccionE>0 && EstadoE>0)
        ReduccionE=ReduccionE-2;
    elseif (ReduccionA>0 && EstadoA>0)
        ReduccionA=ReduccionA-2;
    elseif (ReduccionM>0 && EstadoM>0)
        ReduccionM=ReduccionM-2;
    end
end

```

Figura 3-17. Algoritmo secuencial de reencendido en la función output de la función “modulado_y_apagado.m”

Si la diferencia entre la potencia instantánea total consumida por la casa y la potencia máxima que se ha puesto como límite o la generada en ese momento es menor que 300 W, se deberán empezar a reencender o aumentar de potencia de 20% en 20% aquellos calefactores que hayan sido apagados o reducidos su potencia al tener antes una potencia límite menor. El orden de prioridad es el mismo que en el caso anterior, pero a la inversa. Se aumentará la potencia en los calefactores de mayor prioridad a menor prioridad.

El valor de 300 W se utiliza como coeficiente de seguridad, para no alcanzar la potencia máxima exacta y dejar un margen adecuado.

Una vez ya se han realizado los cambios oportunos a estas variables, sus valores se trasladan del Workspace a Simulink como salidas del bloque de esta función “modulado_y_apagado.m”:

```
%% Escritura salidas
%Reduccion zona A
block.OutputPort(1).Data = double(ReduccionA);
%Reduccion zona D
block.OutputPort(2).Data = double(ReduccionD);
%Reduccion zona E
block.OutputPort(3).Data = double(ReduccionE);
%Reduccion zona G
block.OutputPort(4).Data = double(ReduccionG);
%Reduccion zona H
block.OutputPort(5).Data = double(ReduccionH);
%Reduccion zona I
block.OutputPort(6).Data = double(ReduccionI);
%Reduccion zona J
block.OutputPort(7).Data = double(ReduccionJ);
%Reduccion zona K
block.OutputPort(8).Data = double(ReduccionK);
%Reduccion zona L
block.OutputPort(9).Data = double(ReduccionL);
%Reduccion zona M
block.OutputPort(10).Data = double(ReduccionM);
%Reduccion zona N
block.OutputPort(11).Data = double(ReduccionN);

%endfunction
```

Figura 3-18. Definición de salidas en la función output de la función “modulado_y_apagado.m”

El bloque que ejecuta esta función “modulado_y_apagado.m” en Simulink tiene la siguiente forma:

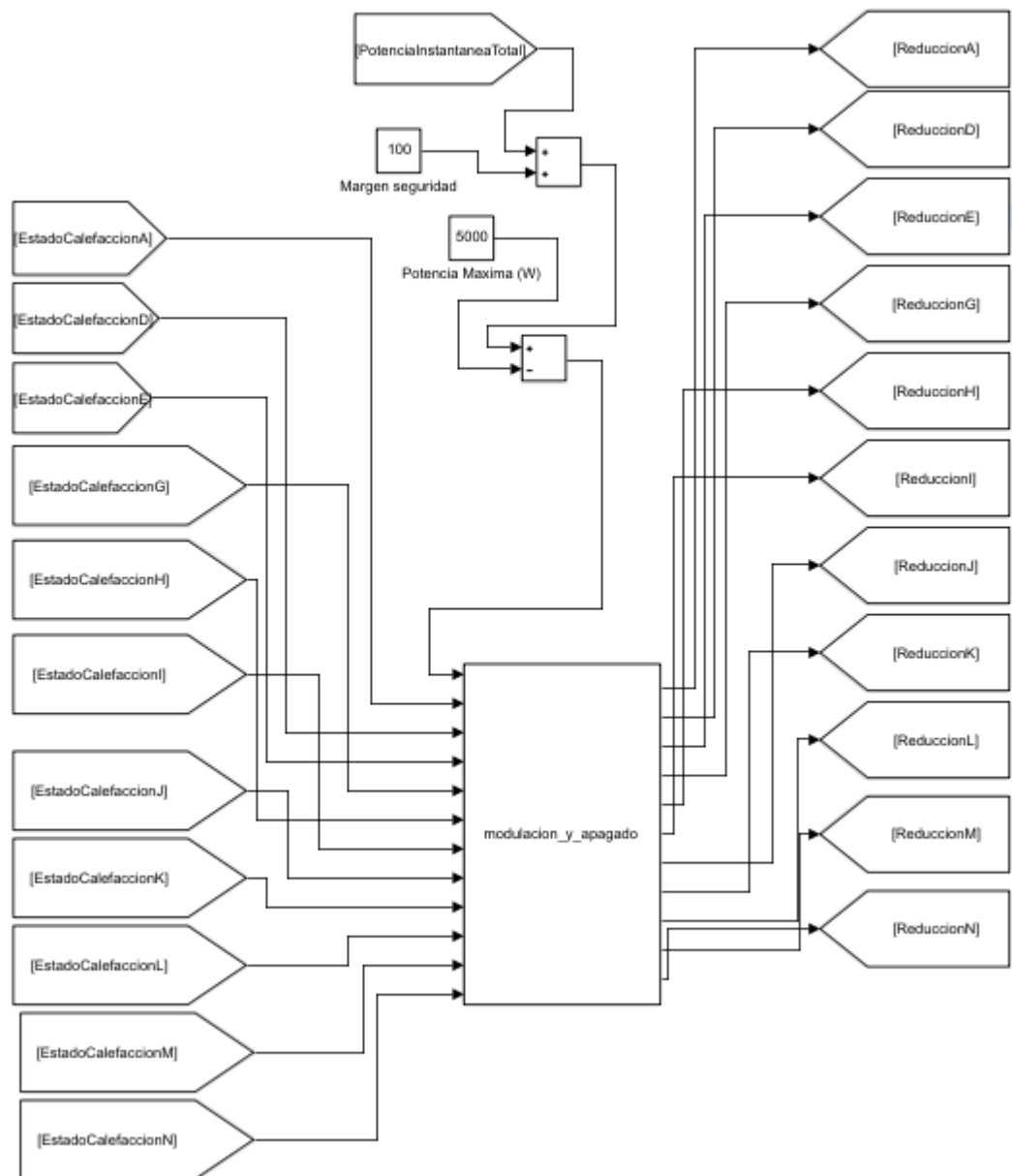


Figura 3-19. Bloque de Simulink que ejecuta la función “modulación_y_apagado.m”

Se ha considerado otro factor de seguridad para dejar cierto margen cuando se reduzca la potencia de calefactores, de forma que la diferencia entre la potencia instantanea total y la potencia máxima tenga una diferencia de mínimo 100 W.

Por último, se escriben los valores correspondientes de tensión de cada calefactory de cada habitación en sus variables de Home I/O, modificando adecuadamente la función “write.m”. El código de la función esta completa en el anexo H. En Simulink, se obtiene lo siguiente:

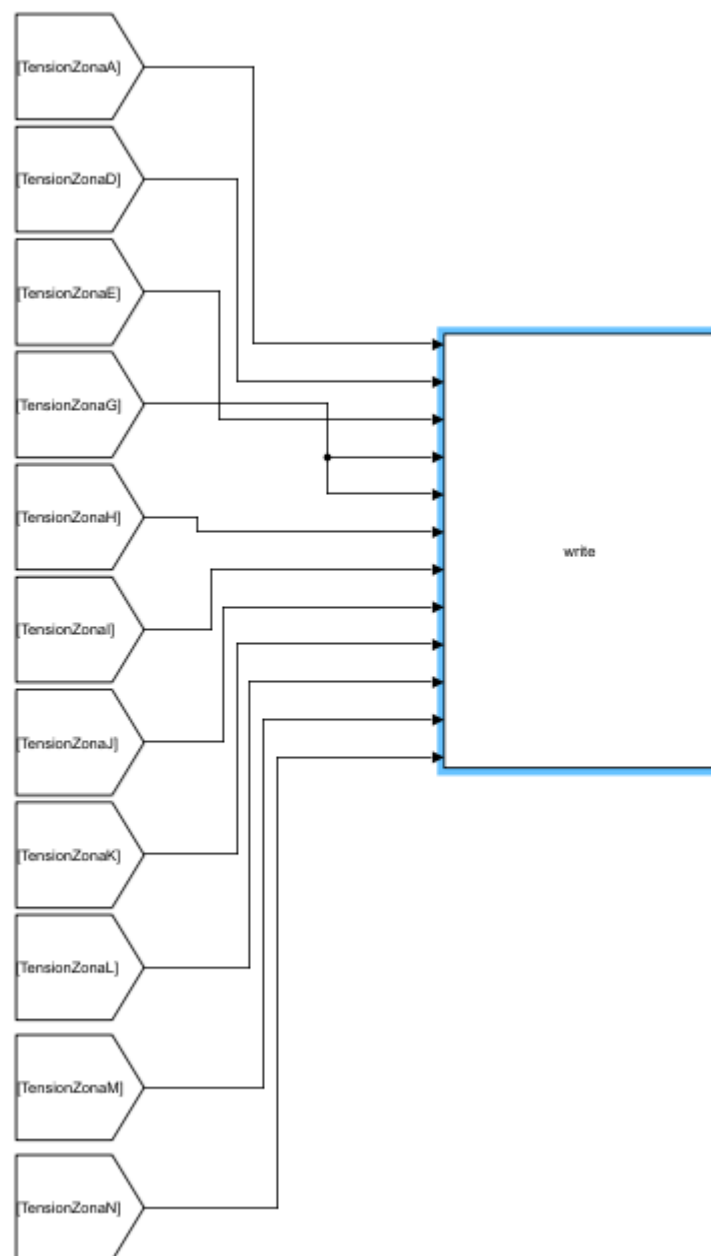


Figura 3-20. Bloque de escritura de las tensiones a todos los calefactores en Simulink

3.3 Control predictivo basado en modelo (solo microrred) para el Sistema A

Este apartado es parecido al anterior, pero con una serie de diferencias. Se controla únicamente el calefactor de la zona H de la vivienda de la misma forma que en el apartado anterior.

Como sistema de alimentación de energía eléctrica se utiliza la microrred, sin incluir el modelo del calefactor. Es decir, se sigue el diagrama de la Figura 2-17. Microrred definitiva simulada en Simulink con su respectivo controlador MPC **¡Error! No se encuentra el origen de la referencia..** Así, se ven las similitudes y diferencias entre un control u otro (incluyendo el modelo del calefactor en el MPC y sin incluirlo). El control del calefactor se realizará mediante un controlador todo/nada con histéresis, como el utilizado en el apartado anterior. En este caso, carece de sentido el utilizar el orden de prioridad, ya que la microrred siempre tendrá capacidad suficiente para alimentar al calefactor, gracias a sus diversas fuentes de alimentación. Por lo que, el

calefactor siempre funcionará a 1 kW cuando esté encendido. El tiempo de muestreo pasa a ser de 10 segundos.

Igualmente, es interesante realizar el ensayo y ver como funciona la microrred en los diferentes casos para el sistema A, con el mismo perfil de generación de energía renovable.

El diagrama en Simulink es el siguiente:

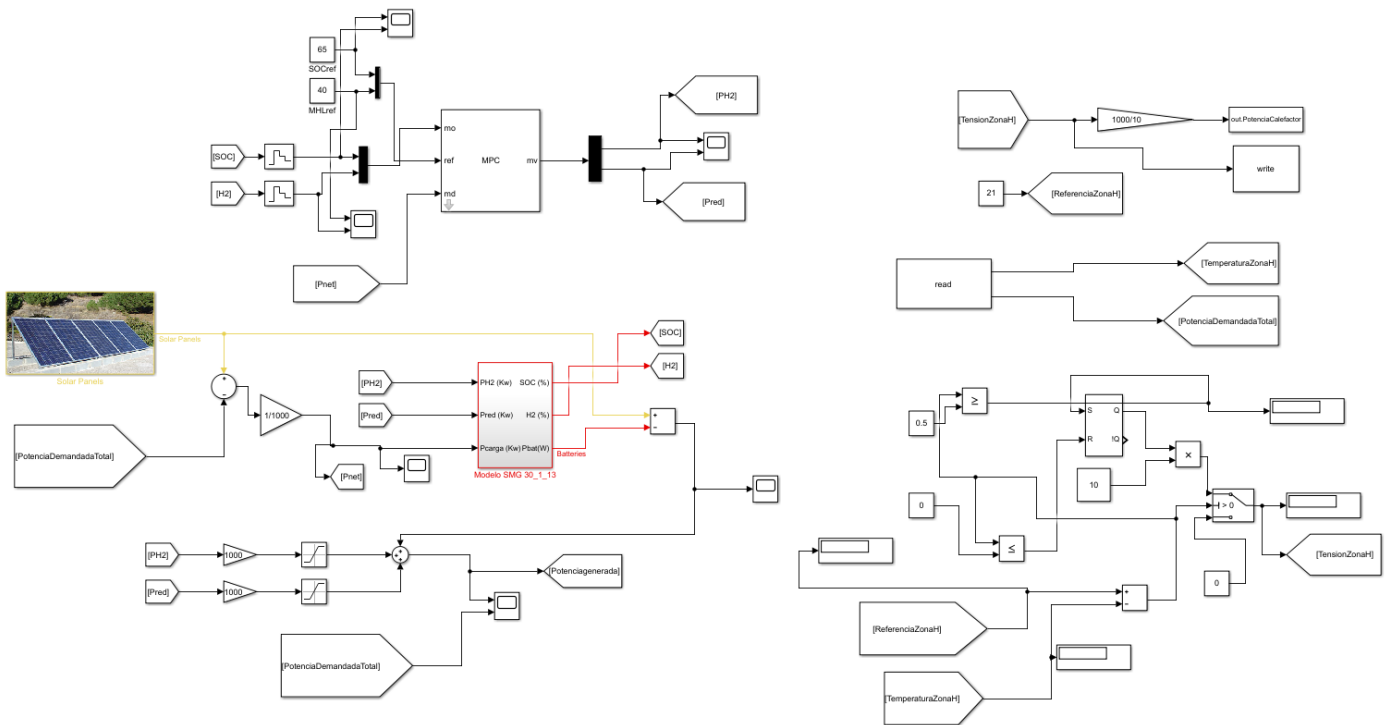


Figura 3-21. Modelo en Simulink del Sistema A con el controlador MPC (solo microrred)

4 ANÁLISIS DE ESTUDIOS

En este capítulo se van a analizar los diferentes controladores y su funcionamiento en las diferentes situaciones que se han estudiado a lo largo de este trabajo.

4.1 Análisis y resultados del Sistema A

4.1.1 Ensayos utilizando el controlador MPC

En este apartado, se van a realizar los ensayos utilizando el diagrama de Simulink de la Figura 3-21.

- Ensayo nº1: los parámetros del controlador MPC de la microred no se han modificado. Son similares a los que aparecen en el apartado 2.3.4 Controlador MPC de la microred. Se obtienen los siguientes

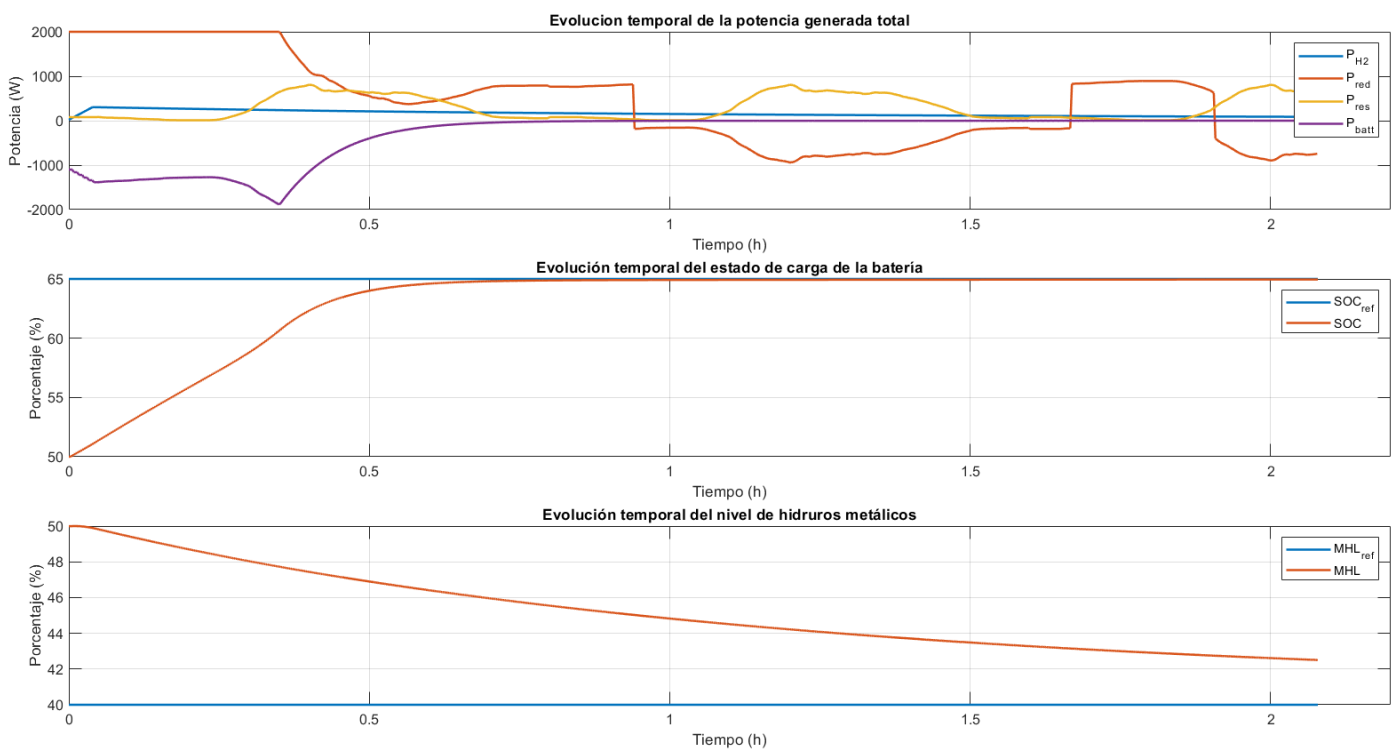


Figura 4-1. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) nº1 sistema A resultados:

Se observa que:

- El electrolizador no trabaja en ningún momento, ya que $P_{H_2}(k)$ toma solamente valores mayores que cero a lo largo de la simulación. Esto implica que solo trabaja la pila de combustible.
- La potencia de la red se usa considerablemente, sobre todo al inicio de la simulación.

- El perfil de generación de los paneles solares se ha obtenido de Internet, gracias a las estadísticas que ofrece el Gobierno de España y se ha escalado para que los valores fueran adecuados para este proyecto. Se usará este mismo perfil para todos los ensayos.
- Las baterías solo absorben energía a lo largo de la simulación. No aportan en ningún momento potencia eléctrica.
- Tanto el estado de carga de la batería como el nivel de hidruros metálicos se acercan adecuadamente a sus respectivas referencias que se habían establecido.

Las diferentes variables tienen este comportamiento debido los pesos que se les han asignado. En los siguientes ensayos, se modificarán los pesos y se observará como se modifican sus comportamientos.

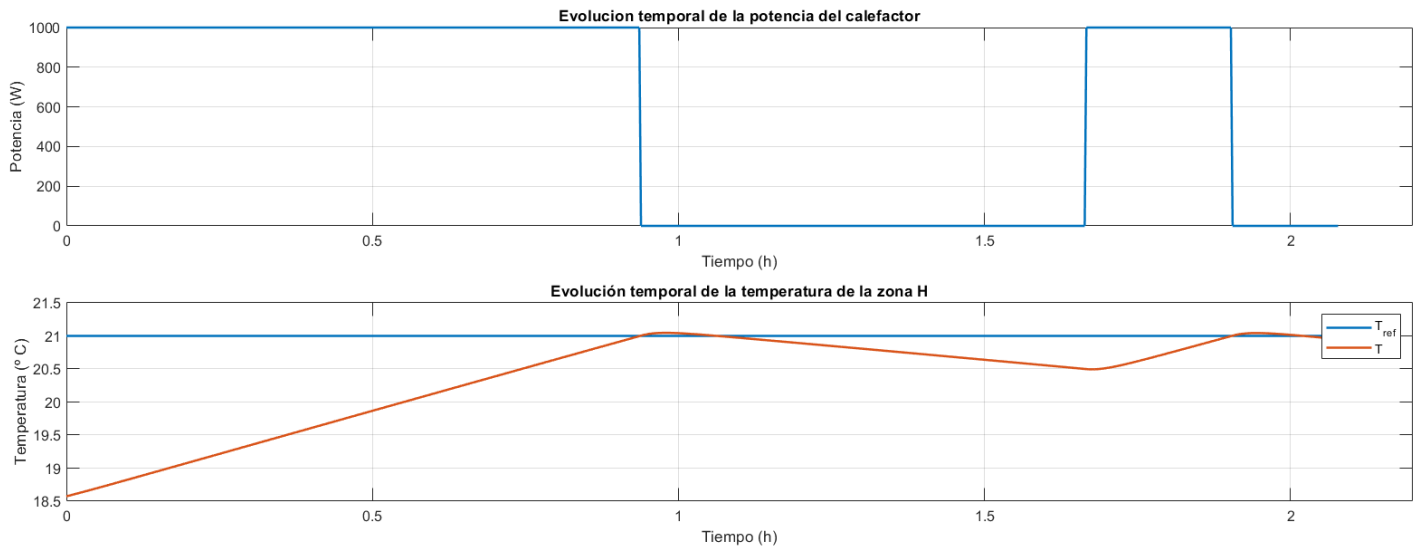


Figura 4-2. Evolución temporal de las variables del calefactor, ensayo MPC (solo microred) n°1 sistema A

En cuanto al control del calefactor, se ve que funciona adecuadamente, como se había diseñado.

- Ensayo n°2: El peso de $P_{grid}(k)$ va a cambiar su valor de 0 a 100 y el peso de $SOC(k)$ de ser 1 a 0.001. Se obtienen los siguientes resultados:

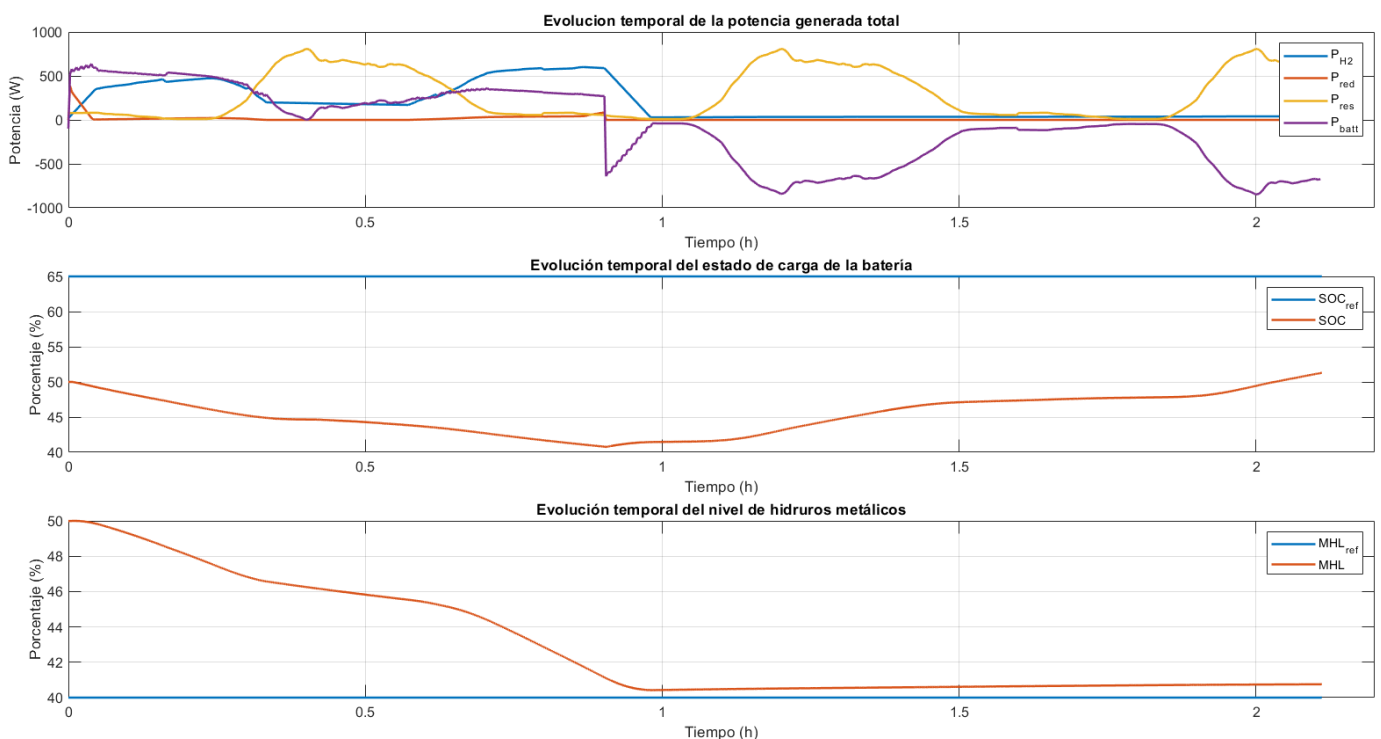


Figura 4-3. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) n°2 sistema A

Se observa que:

- El electrolizador sigue sin funcionar en ningún momento, ya que $P_{H_2}(k)$ toma solamente valores mayores que cero.
- Se ha reducido el uso de la potencia de la red drásticamente. Esto es debido al aumento del peso que le hemos dado a esta variable. Un aumento de peso a una entrada provoca que el controlador de menos prioridad a la hora de utilizarla.
- El perfil de generación de los paneles solares se mantiene similar.
- Las baterías en este caso empiezan generando potencia eléctrica, para luego absorber. Esto se produce ya que el peso del estado de carga de la batería se ha disminuido drásticamente. Una disminución del peso a una salida da lugar a que la prioridad del controlador de llevar dicha variable al valor de referencia sea menor. Esto permite que una mayor utilización de las baterías como consecuencia.
- Como se acaba de mencionar, la prioridad del controlador de llevar el nivel de estado de carga de la batería es mucho menor. Por lo que, el estado de carga se encuentra más alejado de su referencia esta vez. En cambio, el nivel de hidruros metálicos se mantiene como antes, e incluso, se disminuye el tiempo que tarda en acercarse a su valor de referencia. Esto se debe a que el controlador pone todo el esfuerzo en llevar el nivel de hidruros metálicos al valor de referencia, gracias a la disminución del peso del estado de carga de la batería.

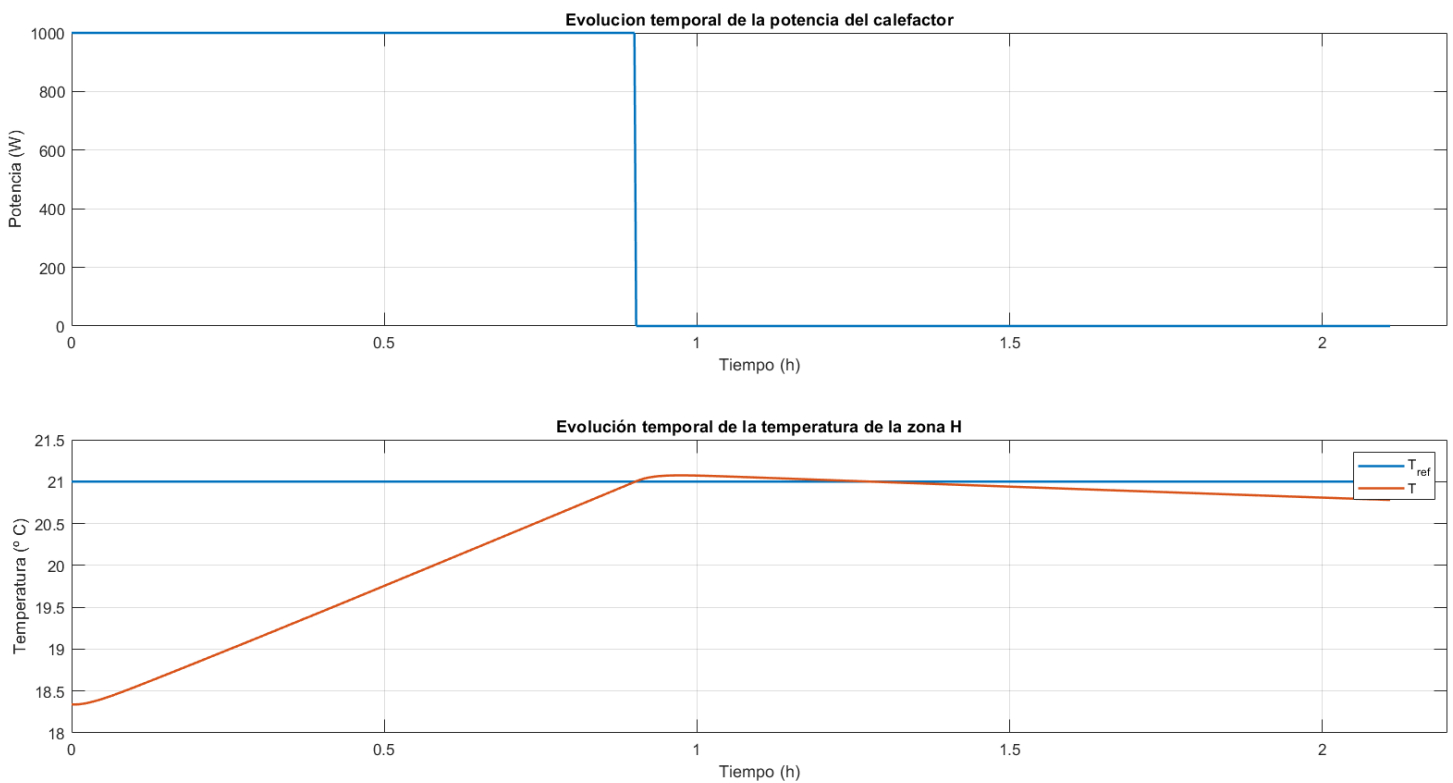


Figura 4-4. Evolución temporal de las variables del calefactor, ensayo MPC (solo microrred) n°2 sistema A

En cuanto al control del calefactor, se observa que sigue funcionando adecuadamente.

- Ensayo n°3: Por último, vamos a mantener las modificaciones de los pesos que se han producido en el apartado anterior, y además, se va a modificar el peso de $MLH(k)$ de 1 a 0.001. Se obtienen los siguientes resultados:

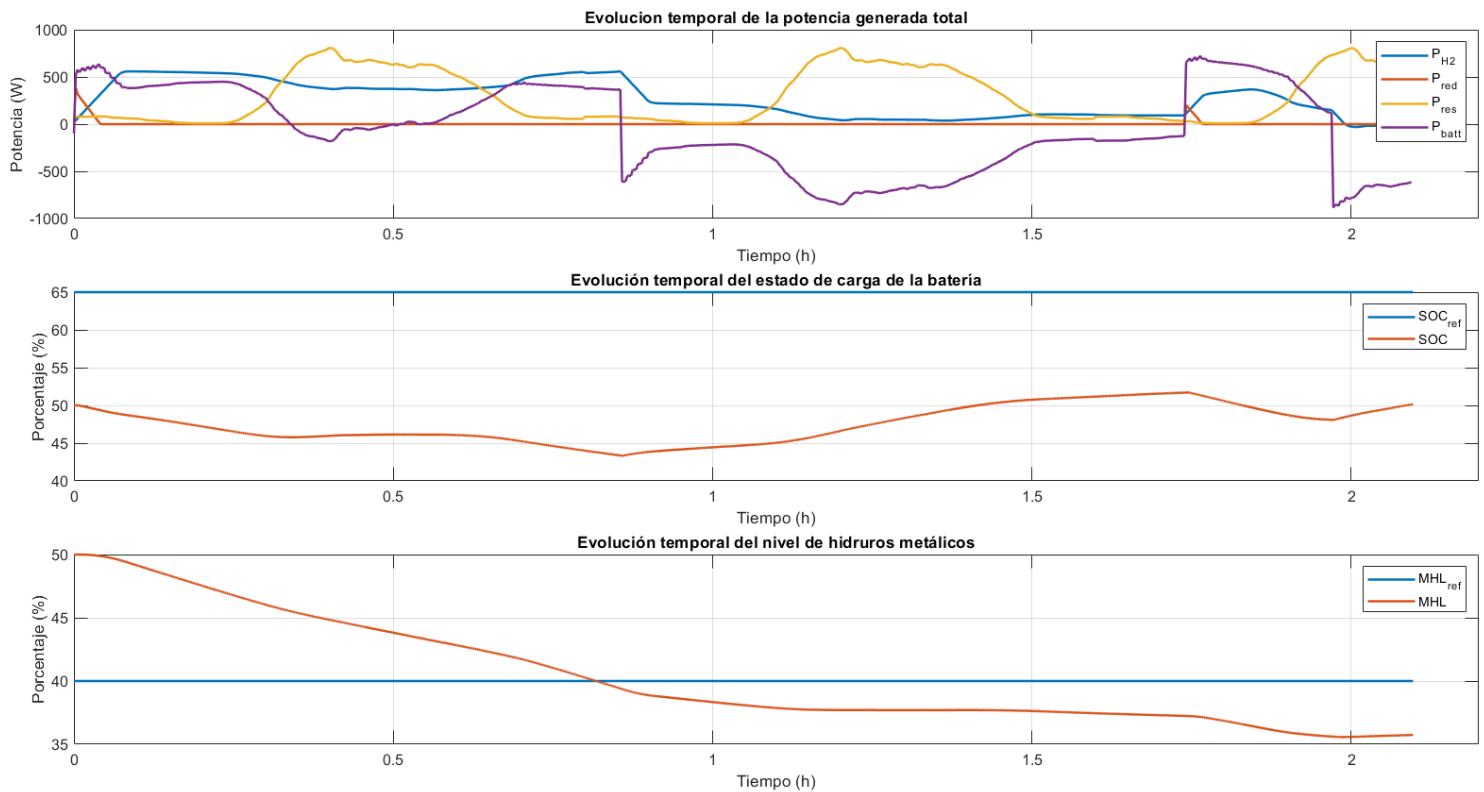


Figura 4-5. Evolución temporal de las variables de la microred, ensayo MPC (solo microred) n°3 sistema A

Se observa que:

- El electrolizador sigue sin funcionar en ningún momento, ya que $P_{H_2}(k)$ toma solamente valores mayores que cero. Pero se observa cierta diferencia con el comportamiento del ensayo anterior. El uso de la pila de combustible se ha visto aumentado considerablemente con respecto al ensayo anterior. Esto ocurre por la disminución del peso del nivel de hidruros metálicos.
- Al igual que en el ensayo anterior, la potencia de la red no se usa prácticamente.
- El perfil de generación de los paneles solares se mantiene similar.
- Las baterías tienen un comportamiento similar.
- El estado de carga de la batería sufre una evolución similar a la del apartado anterior, pero en este caso, cambia la evolución del nivel de hidruros metálicos. Ambas variables de salida tienen un peso de 0.001. Como consecuencia, el esfuerzo del controlador por mantener las variables de salida cerca de su referencia va a ser notablemente menor.

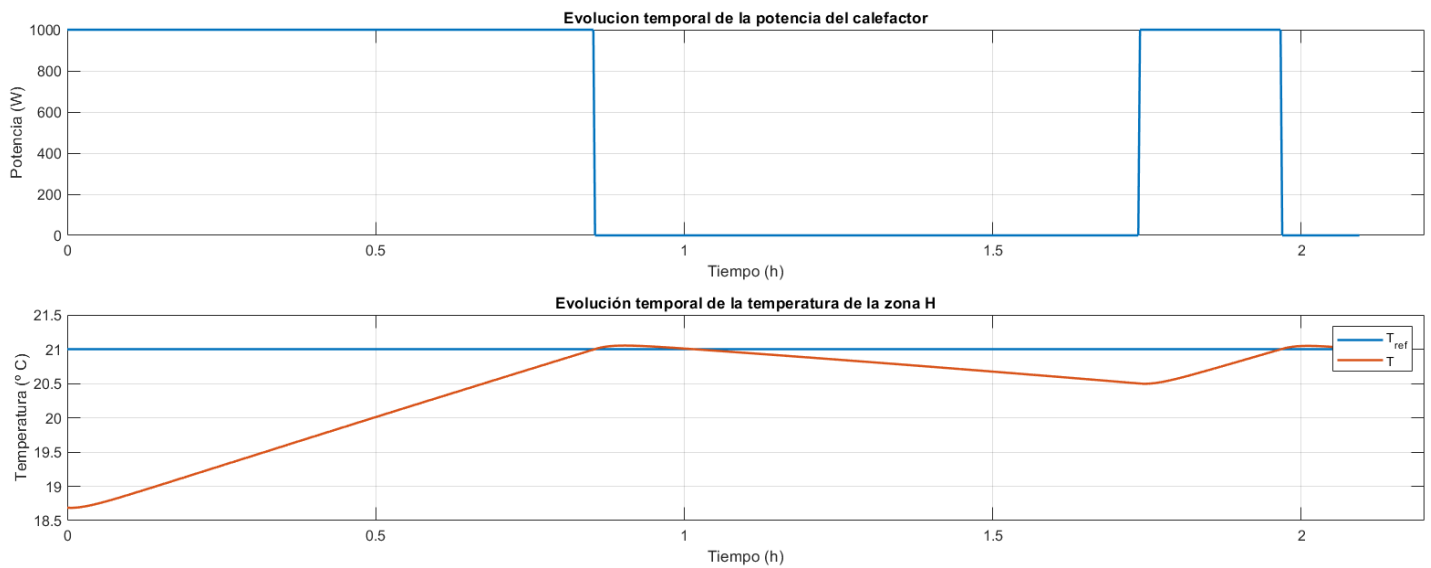


Figura 4-6. Evolución temporal de las variables del calefactor, ensayo MPC (solo microrred) n°3 sistema A

Se observa que en ningún momento estos cambios afectan al correcto funcionamiento del control de la temperatura.

4.2 Análisis y resultados del Sistema B

Para estudiar el correcto funcionamiento del controlador basado en reglas para el **Sistema B**, se van a realizar una serie de ensayos con diferentes condiciones de funcionamiento.

4.2.1 Ensayos con potencia máxima constante

Se realiza un ensayo en el que se encienden los calefactores alfabéticamente de forma ordenada. Primero, se enciende el calefactor de la zona A; segundo, el calefactor de la zona D; tercero, el calefactor de la zona E... y así sucesivamente, hasta encenderse los calefactores de todas las habitaciones. El tiempo entre el encendido de un calefactor y del siguiente es de 10 segundos. La potencia máxima es constante y limitada a 6000 W.

Los resultados del ensayo se pueden observar en la siguiente figura.

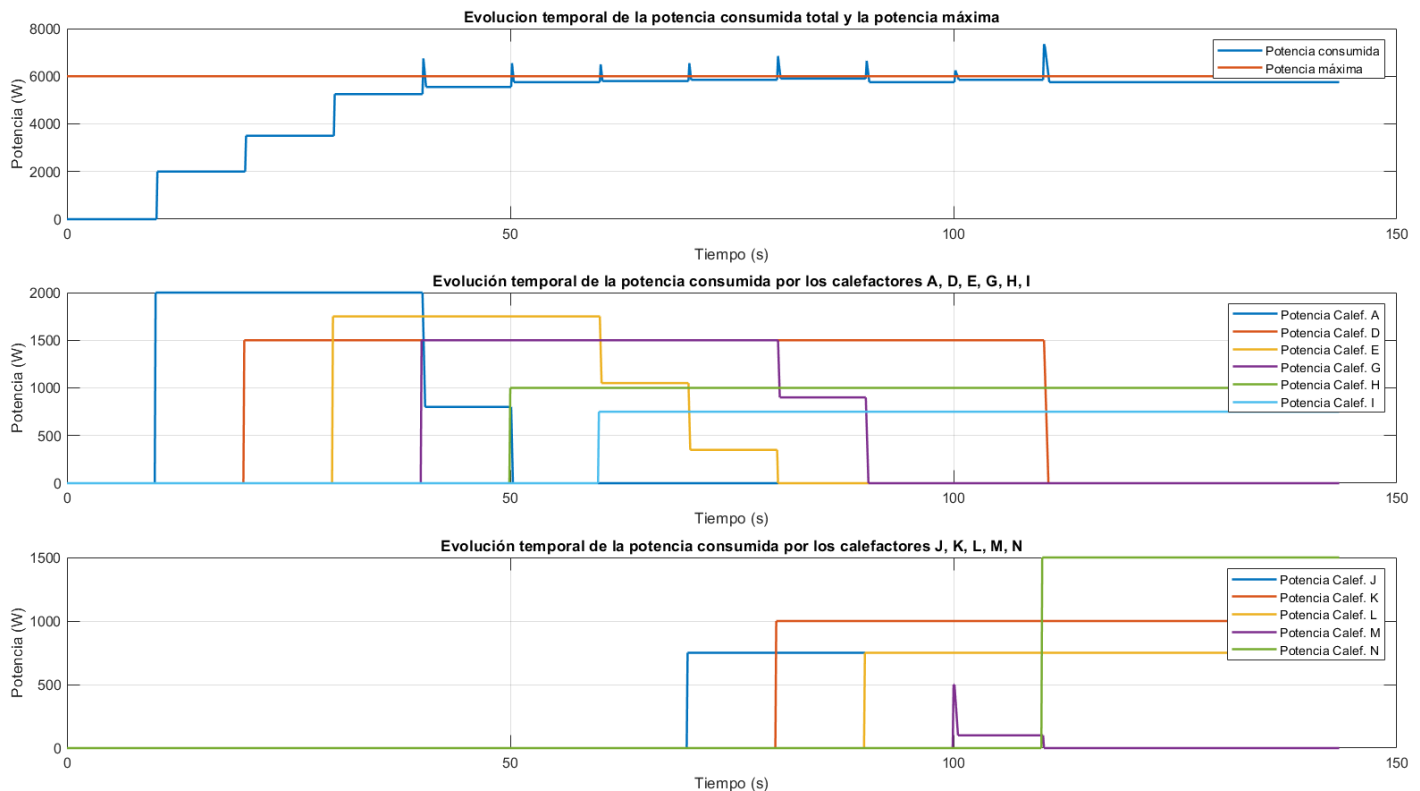


Figura 4-7. Ensayo sistema B con potencia máxima constante encendiendo progresivamente los calefactores

Como se puede observar, se van activando los calefactores 1 a 1 cada 10 segundos. En la potencia total consumida se pueden apreciar una serie de picos una vez que la potencia consumida se acerca a la potencia máxima. Esto es debido a que cuando la potencia consumida alcanza valores cercanos a la potencia máxima, y se active el siguiente calefactor, la potencia consumida superará durante un pequeño instante de tiempo la potencia máxima. Una vez detecta este salto el controlador, actúa consecuentemente, apagando o reduciendo la potencia de los calefactores pertinentes, y la potencia consumida se vuelve a encontrar por debajo de la potencia máxima establecida. El orden de prioridad establecido para el apagado se encuentra en la Figura 3-16.

Para el siguiente ensayo, la situación inicial va a ser la situación final del anterior ensayo, es decir, en el que todos los calefactores están activados, aunque estarán funcionando aquellos que el algoritmo que hemos programado permita. La potencia máxima seguirá constante a 6000 W. En este caso, se apagan uno a uno los calefactores de la habitación de forma ordenada cada 10 segundos de la misma forma que antes, pero en sentido contrario. Es decir, primero, se apaga el calefactor de la habitación N; segundo, se apaga el calefactor de la habitación M; tercero, se apaga el calefactor de la habitación L...

Los resultados de este ensayo se observan en la siguiente figura.

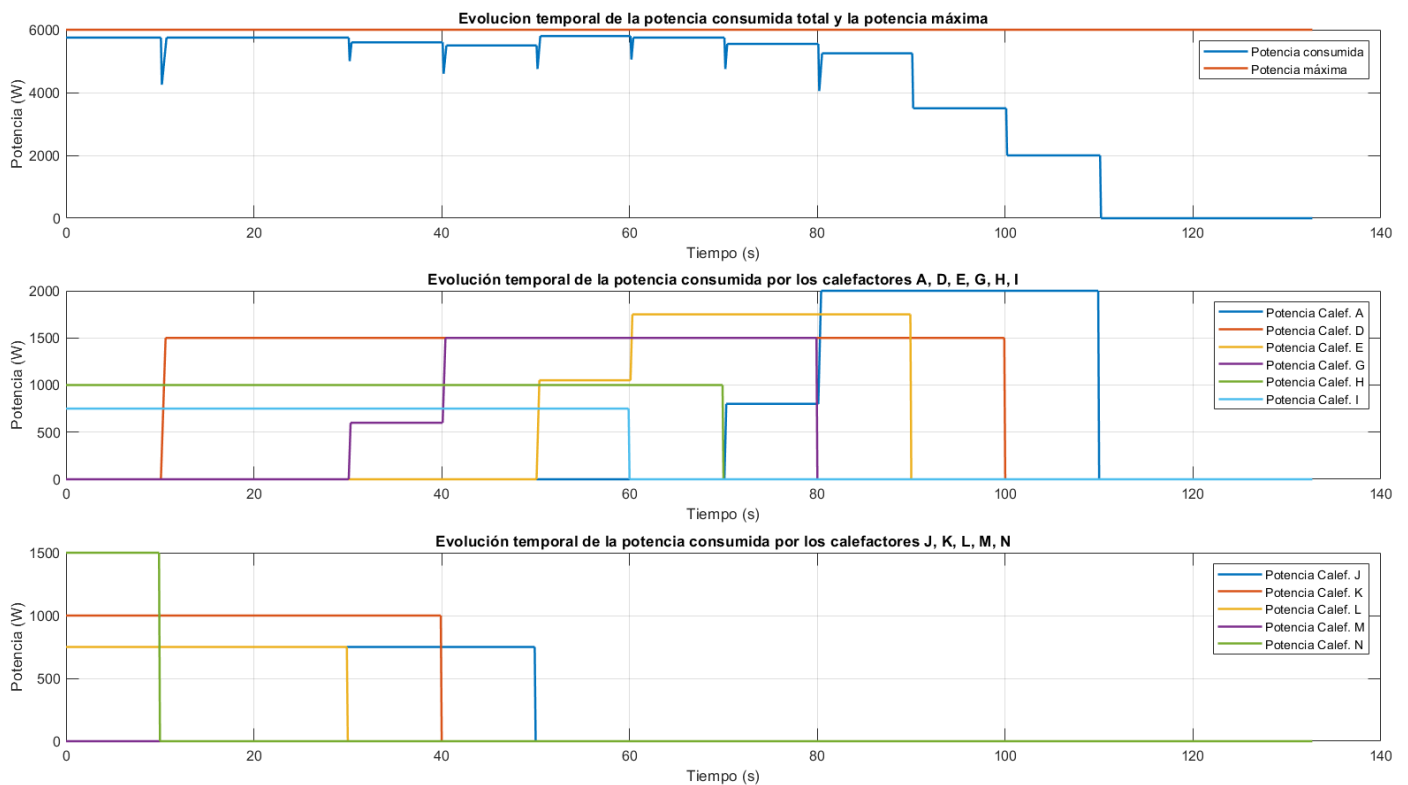


Figura 4-8. Ensayo sistema B con potencia máxima constante apagando progresivamente los calefactores

Como se puede ver, se van desactivando los calefactores 1 a 1 cada 10 segundos. En la potencia total consumida se pueden apreciar una serie de picos, esta vez hacia abajo, cuando la potencia consumida tiene valores cercanos a la potencia máxima. Esto es debido a que cuando se apaga un calefactor, la potencia disminuye considerablemente. Como la potencia disminuye y hay un margen adecuado hasta la potencia máxima, lo que se realiza es un reencendido de los calefactores que han sido apagados anteriormente o han visto reducida su potencia. El orden de prioridad establecido para el reencendido se encuentra en la Figura 3-17.

4.2.2 Ensayo con el número de calefactores activados constante

Se encienden los calefactores de las zonas siguientes: A, E, H, J, L, N. Se parte de una potencia máxima de 10000W; luego, se reduce a 3000 W; después, se aumenta a 5000 W, más tarde; se reduce a 1000 W y, por último, aumenta hasta 7000 W. Los cambios se realizarán cada 15 segundos.

El resultado se observa en la siguiente figura.

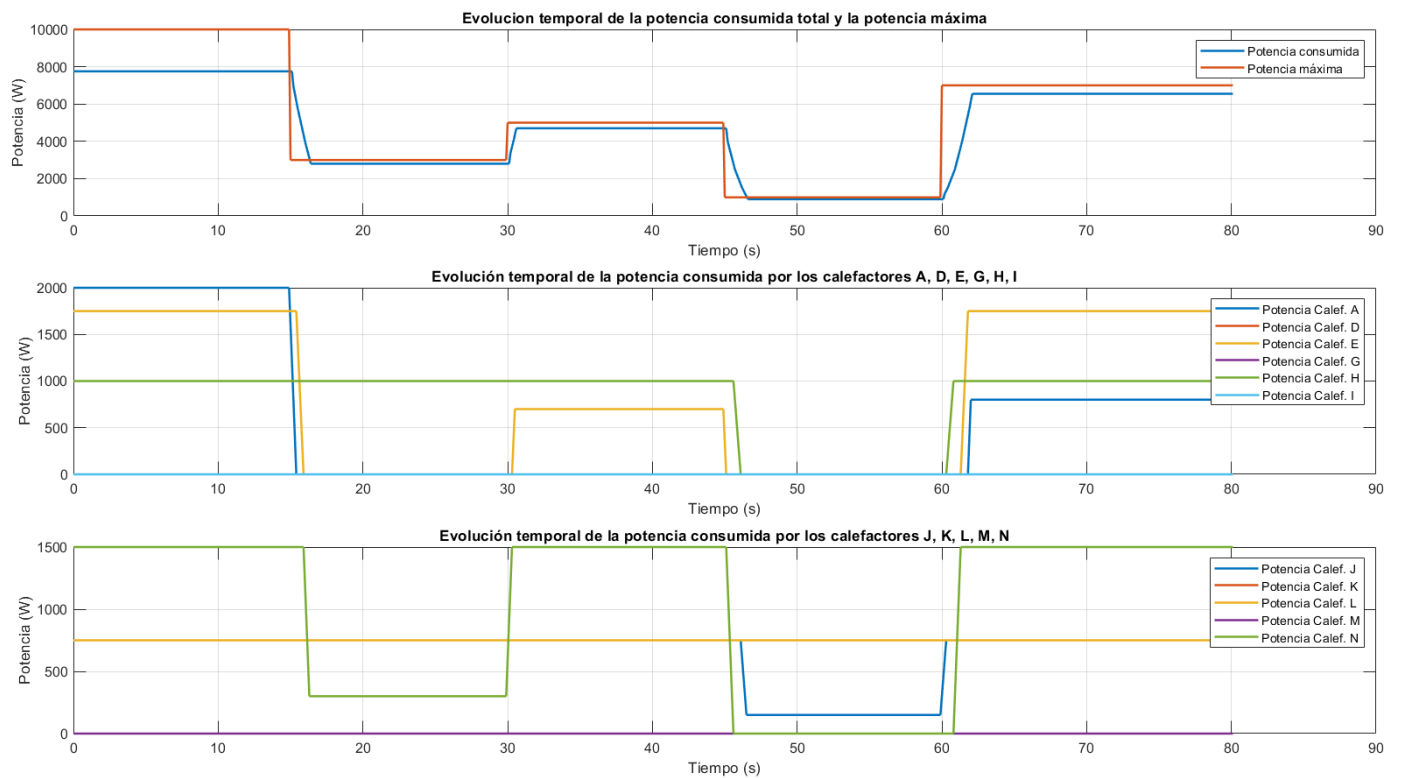


Figura 4-9. Ensayo sistema B con número de calefactores activados constante

Se puede observar como, aunque se modifique la potencia máxima, el controlador siempre va a modificar la potencia consumida total de forma que se mantenga con valores menos que la potencia máxima, independientemente de cual sea el valor de ésta. Para ello, se realizarán los apagados o reencendidos pertinentes según el algoritmo.

4.2.3 Ensayo con ninguna variable constante

Por último, se va a realizar un ensayo en el que ninguna variable se mantiene constante. Tanto la potencia máxima como el número de calefactores activados varía a lo largo de la simulación. El resultado se observa en la siguiente figura.

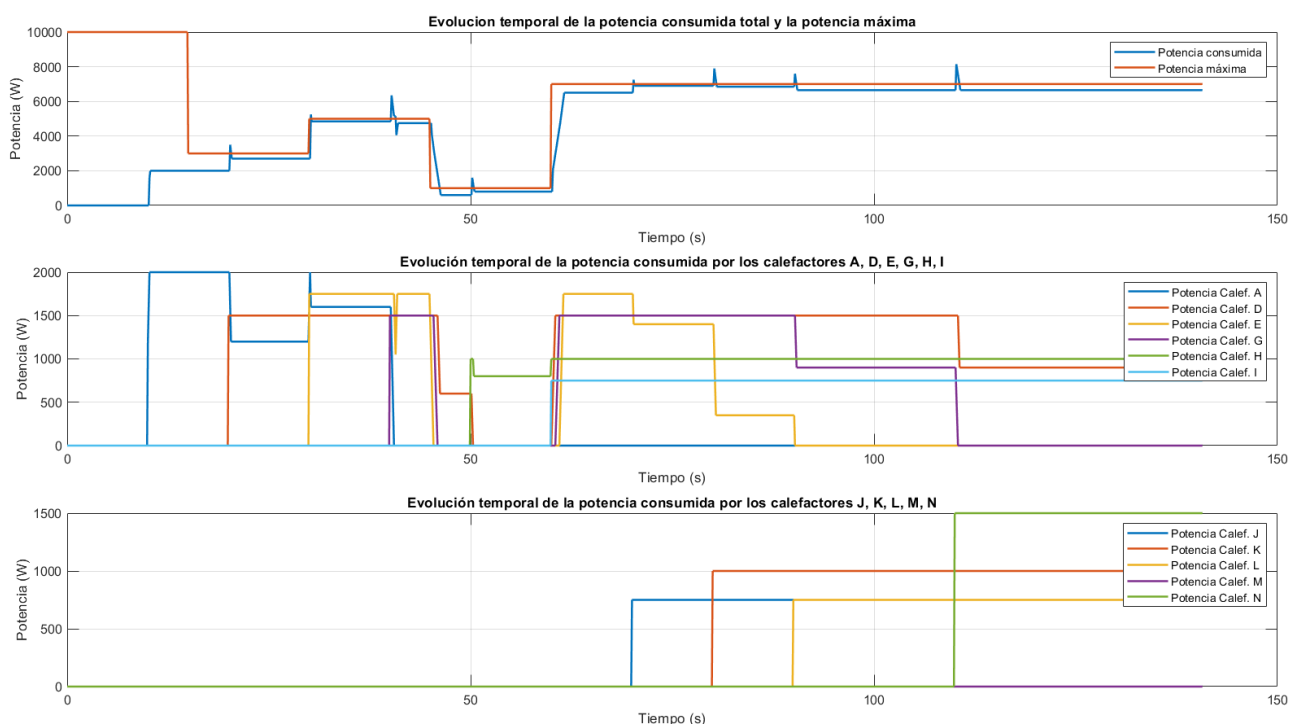


Figura 4-10. Ensayo sistema B con ninguna variable constante

Se puede apreciar que el controlador sigue trabajando adecuadamente y no supone ningún obstáculo.

5 CONCLUSIONES Y TRABAJOS FUTUROS

En este último capítulo se van a explicar las conclusiones a las que se han llegado después de la realización del trabajo, las limitaciones que se han observado y posibles trabajos futuros que continúen la investigación realizada en este trabajo.

5.1 Limitaciones encontradas

A lo largo de la realización de este trabajo, se han encontrado ciertas limitaciones en cuanto al programa de Home I/O. Si bien es cierto que el programa no ha ocasionado ningún obstáculo, hay ciertos aspectos de la vivienda que faltan por incluir para que se asemeje aún más a la realidad.

Por ejemplo, se puede destacar la ausencia de electrodomésticos y tomas de corriente. Estos elementos consumen una potencia eléctrica considerable y, por tanto, afectan al consumo eléctrico destacablemente. La inclusión de lavajillas, lavadora, frigorífico, horno, microondas... Son aparatos eléctricos que se encuentran en cualquier vivienda y cuya existencia se ha echado en falta a la hora de gestionar la energía eléctrica en la realización de este trabajo.

5.2 Conclusiones

Como conclusión, se han obtenido resultados que demuestran la utilidad tanto de utilizar un controlador MPC para el **sistema A** como un control más simple y básico, como puede ser un control basado en reglas para el **sistema B**. Ambos han cumplido con la función que se esperaba y permitirían reducir el consumo eléctrico notablemente en aquellas viviendas en las que se aplicase.

El control basado en reglas es más útil para casos como el que se ha estudiado. Cuando toda la potencia eléctrica viene dada por la red eléctrica pública y no se puede evitar utilizarla como consecuencia. No es necesario ningún proceso de optimización en cuanto a la obtención de energía eléctrica se refiere. Por tanto, aunque sea una solución más simple, puede ser igualmente válida y a considerar a la hora de enfrentar problemas de ahorro de energía eléctrica.

En cambio, el controlador MPC es una forma de controlar más sofisticada y compleja, y resuelve básicamente un problema de optimización según las necesidades del usuario. Este controlador se vuelve muy necesario a la hora de gestionar las diferentes fuentes de energía eléctrica en la red, siendo su uso prácticamente obligado. Permite dar más prioridad a unas fuentes que a otras, y cuya prioridad también se puede modificar cuando el usuario lo requiera modificando los pesos.

5.3 Trabajos futuros

En cuanto a desarrollos futuros, se plantean los siguientes:

- Para el primer caso, aquel en el que se vive en una vivienda cuya energía eléctrica viene toda dada por la red eléctrica pública. La posible solución que se ha propuesto es la de establecer una potencia máxima en la vivienda y establecer un control basado en reglas. Esto es especialmente útil en esta época, ya que actualmente existe una discriminación horaria en cuanto a tarifa de luz se refiere, siendo las siguientes:



Figura 5-1. Horarios de las distintas partes que tiene la tarifa de luz con discriminación horaria [9]

Los diferentes horarios que se han mostrado en la figura anterior corresponden al horario de lunes a viernes laborables. Es importante destacar que la tarifa de la hora valle también se aplica los sábados, domingos y festivos nacionales. Existen estos tres períodos en el que cada uno tiene un precio de euro por MWh diferente, siendo el más caro en la hora punta; después, el de la hora llana; y, por último, el de la hora valle.

Por tanto, se podría realizar un proyecto en el que se estableciese una potencia máxima para cada tramo horario. En la hora valle, esa potencia máxima se podría aumentar considerablemente porque tiene el menor precio y resultaría más barato; en la hora llana considerar un límite de potencia intermedio; y en la hora punta establecer una potencia baja, ya que es el tramo horario con el precio más caro. El control de los diferentes aparatos eléctricos regulables de la vivienda se podría realizar mediante un controlador MPC en vez de utilizar un control basado en reglas. Esto permitiría una solución en la que no se tendrían que tomar soluciones tan drásticas como que algunos calefactores funcionen a su potencia máxima mientras otros se encuentren apagados. Utilizando un controlador MPC, y cambiando sus pesos de manera adecuada, se podría conseguir el mismo objetivo, pero aumentando el confort del usuario. Las entradas al modelo serían las diferentes tensiones de entrada de los calefactores, y las salidas serían las diferentes temperaturas de cada habitación. Así, haciendo una distribución de pesos adecuada, se podría conseguir un mejor control de la potencia de los diferentes calefactores.

- Para el segundo caso, aquel en el que la vivienda posee una microred como fuente de alimentación de energía eléctrica. En este caso, la posible solución que se ha propuesto ha sido solo considerando el calefactor de la zona H por falta de tiempo para la investigación, ya que todo el trabajo realizado ha sido prácticamente nuevo y para el cual ha sido necesario informarse de muchos temas específicos.

Entonces, esto abre la posibilidad de continuar con este trabajo y considerar todos los calefactores de la vivienda. Se podría tener un controlador MPC conjunto, que incluyese la microred y todos los calefactores de la vivienda. Para ello, haría falta construir un nuevo modelo, que se describe a

continuación. Las entradas al modelo serían las diferentes tensiones de entrada de los calefactores, y las salidas serían las diferentes temperaturas de cada habitación. Una vez obtenido el modelo, se podría añadir al modelo de la microred. De esta forma, podrían obtenerse resultados destacables de esta continuación del trabajo. Así, tendríamos tanto la generación de potencia optimizada, como la potencia que consume cada calefactor. Esto conllevaría un gran ahorro energético, y como consecuencia, un menor coste en la factura la luz, tan necesaria en la etapa actual.

REFERENCIAS

- [1 Real Games, «Home I/O,» [En línea]. Available: <https://docs.realgames.co/homeio/en/>.
]
- [2 P. Velarde, L. V. Isorna, J. M. M. Torreblanca, C. Ocampo-Martínez y C. B. Alba, «On the Comparison of Stochastic Model Predictive Control Strategies Applied to a Hydrogen-based Microgrid,» *Journal of power sources*, vol. 343, pp. 161-173, Marzo 2017.
- [3 M. Retuerto, «ideagreen.es,» [En línea]. Available: <https://ideagreen.es/hidrogeno-verde/tipos-de-electrolizadores/>.
]
- [4 «Wikipedia,» [En línea]. Available: [https://es.wikipedia.org/wiki/Pila_de_combustible#Pila_de_Combustible_de_Membrana_de_intercambio_prot%C3%B3nico_\(PEM\)](https://es.wikipedia.org/wiki/Pila_de_combustible#Pila_de_Combustible_de_Membrana_de_intercambio_prot%C3%B3nico_(PEM)).
]
- [5 Mathworks, «Mathworks,» [En línea]. Available: <https://uk.mathworks.com/help/mpc/>.
]
- [6 Mathworks, «Mathworks,» [En línea]. Available: <https://uk.mathworks.com/help/ident/>.
]
- [7 R. Fernández, «Statista,» 10 Septiembre 2021. [En línea]. Available: <https://es.statista.com/estadisticas/993787/precio-medio-final-de-la-electricidad-en-espana/>. [Último acceso: 11 Septiembre 2021].
- [8 M. Á. Noceda, «El país,» 10 Septiembre 2021. [En línea]. Available: <https://elpais.com/economia/2021-09-10/el-precio-de-la-luz-seguira-disparada-este-sabado-por-encima-de-los-150-euros-megavatio-hora.html>. [Último acceso: 11 Septiembre 2021].
- [9 E. Pérez, «Xataka,» 6 Mayo 2021. [En línea]. Available: <https://www.xataka.com/energia/nueva-factura-luz-llega-junio-que-cambia-como-deberemos-adaptar-nuestro-consumo-para-beneficiarnos>.
]

ANEXOS

ANEXO A: Direcciones de memoria de todos los **dispositivos de entrada** del programa HOME I/O

Tipo de dato	Dirección	Zona	Nombre	Tipo de contacto
Bool	0	A	Light Switch 1	NO
Bool	1	A	Light Switch 2	NO
Bool	2	A	Light Switch 3	NO
Bool	3	A	Up/Down Switch 1 (Up)	NO
Bool	4	A	Up/Down Switch 1 (Down)	NO
Bool	5	A	Up/Down Switch 2 (Up)	NO
Bool	6	A	Up/Down Switch 2 (Down)	NO
Bool	7	A	Light Switch Dimmer 1 (Up)	NO
Bool	8	A	Light Switch Dimmer 1 (Down)	NO
Bool	9	A	Light Switch Dimmer 2 (Up)	NO
Bool	10	A	Light Switch Dimmer 2 (Down)	NO
Bool	11	A	Light Switch Dimmer 3 (Up)	NO
Bool	12	A	Light Switch Dimmer 3 (Down)	NO
Bool	13	A	Door Detector 1	NO
Bool	14	A	Door Detector 2	NO
Bool	15	A	Motion Detector	NO
Bool	16	A	Brightness Sensor	NC
Bool	17	A	Smoke Detector	NO
Bool	27	B	Light Switch 1	NO
Bool	28	B	Light Switch 2	NO
Bool	29	B	Motion Detector	NO
Bool	39	C	Light Switch	NO
Bool	49	D	Up/Down Switch (Up)	NO
Bool	50	D	Up/Down Switch (Down)	NO
Bool	51	D	Light Switch Dimmer 1 (Up)	NO
Bool	52	D	Light Switch Dimmer 1 (Down)	NO
Bool	53	D	Light Switch Dimmer 2 (Up)	NO
Bool	54	D	Light Switch Dimmer 2 (Down)	NO
Bool	55	D	Door Detector	NO
Bool	56	D	Motion Detector	NO
Bool	57	D	Brightness Sensor	NC
Bool	67	E	Light Switch 1	NO
Bool	68	E	Light Switch 2	NO
Bool	69	E	Light Switch 3	NO
Bool	70	E	Light Switch 4	NO
Bool	71	E	Light Switch 5	NO
Bool	72	E	Up/Down Switch (Up)	NO
Bool	73	E	Up/Down Switch (Down)	NO
Bool	74	E	Light Switch Dimmer 1 (Up)	NO
Bool	75	E	Light Switch Dimmer 1 (Down)	NO
Bool	76	E	Light Switch Dimmer 2 (Up)	NO

Bool	77	E	Light Switch Dimmer 2 (Down)	NO
Bool	78	E	Door Detector 1	NO
Bool	79	E	Door Detector 2	NO
Bool	80	E	Motion Detector	NO
Bool	81	E	Brightness Sensor	NC
Bool	82	E	Alarm Key Pad (Armed)	NO
Bool	92	F	Light Switch 1	NO
Bool	93	F	Light Switch 2	NO
Bool	94	F	Up/Down Switch 1 (Up)	NO
Bool	95	F	Up/Down Switch 1 (Down)	NO
Bool	96	F	Up/Down Switch 2 (Up)	NO
Bool	97	F	Up/Down Switch 2 (Down)	NO
Bool	98	F	Motion Detector	NO
Bool	99	F	Brightness Sensor	NO
Bool	100	F	Garage Door (Opened)	NO
Bool	101	F	Garage Door (Closed)	NO
Bool	102	F	Garage Door (Infrared)	NO
Bool	112	G	Light Switch 1	NO
Bool	113	G	Light Switch 2	NO
Bool	114	G	Light Switch 3	NO
Bool	115	G	Light Switch 4	NO
Bool	116	G	Light Switch 5	NO
Bool	117	G	Up/Down Switch (Up)	NO
Bool	118	G	Up/Down Switch (Down)	NO
Bool	119	G	Motion Detector	NO
Bool	120	G	Smoke Detector	NO
Bool	130	H	Light Switch 1	NO
Bool	131	H	Light Switch 2	NO
Bool	132	H	Up/Down Switch (Up)	NO
Bool	133	H	Up/Down Switch (Down)	NO
Bool	134	H	Light Switch Dimmer (Up)	NO
Bool	135	H	Light Switch Dimmer (Down)	NO
Bool	136	H	Door Detector 1	NO
Bool	137	H	Door Detector 2	NO
Bool	138	H	Motion Detector	NO
Bool	139	H	Brightness Sensor	NC
Bool	140	H	Smoke Detector	NO
Bool	150	I	Light Switch	NO
Bool	151	I	Light Switch Dimmer (Up)	NO
Bool	152	I	Light Switch Dimmer (Down)	NO
Bool	153	I	Motion Detector	NO
Bool	163	J	Light Switch 1	NO
Bool	164	J	Light Switch 2	NO
Bool	165	J	Up/Down Switch (Up)	NO
Bool	166	J	Up/Down Switch (Down)	NO
Bool	167	J	Light Switch Dimmer (Up)	NO
Bool	168	J	Light Switch Dimmer (Down)	NO
Bool	169	J	Door Detector 1	NO
Bool	170	J	Door Detector 2	NO

Bool	171	J	Motion Detector	NO
Bool	172	J	Brightness Sensor	NC
Bool	173	J	Smoke Detector	NO
Bool	183	K	Light Switch Dimmer (Up)	NO
Bool	184	K	Light Switch Dimmer (Down)	NO
Bool	185	K	Motion Detector	NO
Bool	195	L	Light Switch 1	NO
Bool	196	L	Light Switch 2	NO
Bool	197	L	Up/Down Switch (Up)	NO
Bool	198	L	Up/Down Switch (Down)	NO
Bool	199	L	Light Switch Dimmer 1 (Up)	NO
Bool	200	L	Light Switch Dimmer 1 (Down)	NO
Bool	201	L	Light Switch Dimmer 2 (Up)	NO
Bool	202	L	Light Switch Dimmer 2 (Down)	NO
Bool	203	L	Door Detector 1	NO
Bool	204	L	Door Detector 2	NO
Bool	205	L	Motion Detector	NO
Bool	206	L	Brightness Sensor	NC
Bool	207	L	Smoke Detector	NO
Bool	217	M	Light Switch	NO
Bool	218	M	Up/Down Switch (Up)	NO
Bool	219	M	Up/Down Switch (Down)	NO
Bool	220	M	Door Detector 1	NO
Bool	221	M	Door Detector 2	NO
Bool	222	M	Motion Detector	NO
Bool	223	M	Brightness Sensor	NC
Bool	233	N	Light Switch 1	NO
Bool	234	N	Light Switch 2	NO
Bool	235	N	Light Switch 3	NO
Bool	236	N	Light Switch Dimmer 1 (Up)	NO
Bool	237	N	Light Switch Dimmer 1 (Down)	NO
Bool	238	N	Light Switch Dimmer 2 (Up)	NO
Bool	239	N	Light Switch Dimmer 2 (Down)	NO
Bool	240	N	Light Switch Dimmer 3 (Up)	NO
Bool	241	N	Light Switch Dimmer 3 (Down)	NO
Bool	242	N	Up/Down Switch (Up)	NO
Bool	243	N	Up/Down Switch (Down)	NO
Bool	244	N	Door Detector 1	NO
Bool	245	N	Door Detector 2	NO
Bool	246	N	Motion Detector	NO
Bool	247	N	Brightness Sensor	NC
Bool	248	N	Smoke Detector	NO
Bool	258	O	Motion Detector	NO
Bool	259	O	Brightness Sensor	NC
Bool	260	O	Entrance Gate (Opened)	NO
Bool	261	O	Entrance Gate (Closed)	NO
Bool	262	O	Entrance Gate (Infrared 1)	NO
Bool	263	O	Entrance Gate (Infrared 2)	NO
Bool	264	O	Entrance Gate (Infrared 3)	NO
Bool	274	-	Remote Button 1	NO

Bool	275	-	Remote Button 2	NO
Bool	276	-	Remote Button 3	NO
Bool	277	-	Remote Button 4	NO
Bool	278	-	Remote Button 5	NO
Bool	279	-	Remote Button 6	NO
Bool	280	-	Remote Button 7	NO
Bool	281	-	Remote Button 8	NO
Float	0	A	Brightness Sensor (Analogue)	-
Float	1	A	Thermostat (Room Temperature)	-
Float	2	A	Thermostat (Set Point)	-
Float	3	A	Roller Shades (Openness)	-
Float	4	A	Roller Shades (Openness)	-
Float	5	A	Roller Shades (Openness)	-
Float	6	A	Roller Shades (Openness)	-
Float	12	D	Brightness Sensor (Analogue)	-
Float	13	D	Thermostat (Room Temperature)	-
Float	14	D	Thermostat (Set Point)	-
Float	15	D	Roller Shades (Openness)	-
Float	24	E	Brightness Sensor (Analogue)	-
Float	25	E	Thermostat (Room Temperature)	-
Float	26	E	Thermostat (Set Point)	-
Float	27	E	Roller Shades (Openness)	-
Float	36	F	Brightness Sensor (Analogue)	-
Float	37	F	Roller Shades (Openness)	-
Float	46	G	Thermostat (Room Temperature)	-
Float	47	G	Thermostat (Set Point)	-
Float	48	G	Roller Shades (Openness)	-
Float	57	H	Brightness Sensor (Analogue)	-
Float	58	H	Thermostat (Room Temperature)	-
Float	59	H	Thermostat (Set Point)	-
Float	60	H	Roller Shades (Openness)	-
Float	69	I	Thermostat (Room Temperature)	-
Float	70	I	Thermostat (Set Point)	-
Float	80	J	Brightness Sensor (Analogue)	-
Float	81	J	Thermostat (Room Temperature)	-
Float	82	J	Thermostat (Set Point)	-
Float	83	J	Roller Shades (Openness)	-
Float	92	K	Thermostat (Room Temperature)	-
Float	93	K	Thermostat (Set Point)	-
Float	103	L	Brightness Sensor (Analogue)	-
Float	104	L	Thermostat (Room Temperature)	-
Float	105	L	Thermostat (Set Point)	-
Float	106	L	Roller Shades (Openness)	-
Float	115	M	Brightness Sensor (Analogue)	-
Float	116	M	Thermostat (Room Temperature)	-
Float	117	M	Thermostat (Set Point)	-
Float	118	M	Roller Shades (Openness)	-
Float	127	N	Brightness Sensor (Analogue)	-
Float	128	N	Thermostat (Room Temperature)	-

Float	129	N	Thermostat (Set Point)	-
Float	130	N	Roller Shades (Openness)	-
Float	139	O	Brightness Sensor (Analogue)	-
DateTime	0	-	Date and Time	-

ANEXO B: Direcciones de memoria de todos los **dispositivos de salida** del programa HOME I/O

Tipo de dato	Dirección	Zona	Nombre
Bool	0	A	Lights
Float	0	A	Lights (Analogue)
Bool	1	A	Roller Shades 1 (Up)
Float	1	A	Heater (Analogue)
Bool	2	A	Roller Shades 1 (Down)
Bool	3	A	Roller Shades 2 (Up)
Bool	4	A	Roller Shades 2 (Down)
Bool	5	A	Roller Shades 3 (Up)
Bool	6	A	Roller Shades 3 (Down)
Bool	7	A	Roller Shades 4 (Up)
Bool	8	A	Roller Shades 4 (Down)
Bool	9	A	Heater
Float	11	B	Lights 1 (Analogue)
Float	12	B	Lights 2 (Analogue)
Bool	19	B	Lights 1
Bool	20	B	Lights 2
Float	22	C	Lights (Analogue)
Bool	30	C	Lights
Float	32	D	Lights 1 (Analogue)
Float	33	D	Lights 2 (Analogue)
Float	34	D	Heater (Analogue)
Bool	40	D	Lights 1
Bool	41	D	Lights 2
Bool	42	D	Roller Shades (Up)
Bool	43	D	Roller Shades (Down)
Bool	44	D	Heater
Float	44	E	Lights (Analogue)
Float	45	E	Heater (Analogue)
Bool	54	E	Lights
Bool	55	E	Roller Shades (Up)
Float	55	F	Lights 1 (Analogue)
Bool	56	E	Roller Shades (Down)
Float	56	F	Lights 2 (Analogue)
Bool	57	E	Heater
Bool	58	E	Siren
Bool	59	E	Alarm Key Pad (Arm)
Bool	60	E	Alarm Key Pad (Disarm)
Float	66	G	Lights (Analogue)
Float	67	G	Heater 1 (Analogue)
Bool	68	F	Lights 1

Float	68	G	Heater 2 (Analogue)
Bool	69	F	Lights 2
Bool	70	F	Roller Shades (Up)
Bool	71	F	Roller Shades (Down)
Bool	72	F	Garage Door (Open)
Bool	73	F	Garage Door (Close)
Float	78	H	Lights (Analogue)
Float	79	H	Heater (Analogue)
Bool	83	G	Lights
Bool	84	G	Roller Shades (Up)
Bool	85	G	Roller Shades (Down)
Bool	86	G	Heater 1
Bool	87	G	Heater 2
Float	89	I	Lights 1 (Analogue)
Float	90	I	Lights 2 (Analogue)
Float	91	I	Heater (Analogue)
Bool	97	H	Lights
Bool	98	H	Roller Shades (Up)
Bool	99	H	Roller Shades (Down)
Bool	100	H	Heater
Float	101	J	Lights (Analogue)
Float	102	J	Heater (Analogue)
Bool	110	I	Lights 1
Bool	111	I	Lights 2
Bool	112	I	Heater
Float	112	K	Lights (Analogue)
Float	113	K	Heater (Analogue)
Bool	122	J	Lights
Bool	123	J	Roller Shades (Up)
Float	123	L	Lights (Analogue)
Bool	124	J	Roller Shades (Down)
Float	124	L	Heater (Analogue)
Bool	125	J	Heater
Float	134	M	Lights (Analogue)
Bool	135	K	Lights
Float	135	M	Heater (Analogue)
Bool	136	K	Heater
Float	145	N	Lights 1 (Analogue)
Bool	146	L	Lights
Float	146	N	Lights 2 (Analogue)
Bool	147	L	Roller Shades (Up)
Float	147	N	Lights 3 (Analogue)
Bool	148	L	Roller Shades (Down)
Float	148	N	Heater (Analogue)
Bool	149	L	Heater
Float	158	O	Lights Poarch 1 (Analogue)
Bool	159	M	Lights
Float	159	O	Lights Poarch 2 (Analogue)
Bool	160	M	Roller Shades (Up)

Float	160	O	Lights Pool (Analogue)
Bool	161	M	Roller Shades (Down)
Float	161	O	Lights Garden (Analogue)
Bool	162	M	Heater
Float	162	O	Lights Entrance (Analogue)
Bool	172	N	Lights 1
Bool	173	N	Lights 2
Bool	174	N	Lights 3
Bool	175	N	Roller Shades (Up)
Bool	176	N	Roller Shades (Down)
Bool	177	N	Heater
Bool	187	O	Lights Porch 1
Bool	188	O	Lights Porch 2
Bool	189	O	Lights Pool
Bool	190	O	Lights Garden
Bool	191	O	Lights Entrance
Bool	192	O	Siren
Bool	193	O	Entrance Gate (Open)
Bool	194	O	Entrance Gate (Close)

ANEXO C: Direcciones de memoria de **variables de memoria que solo se pueden leer (no modificar)** del programa HOME I/O

Tipo de dato	Dirección	Zona	Nombre
Bool	256	-	Using DLST
Float	129	-	Time Scalev
Float	130	-	Latitude
Float	131	-	Longitude
Float	132	-	Air Temperature [K]
Float	133	-	Relative Humidity
Float	134	-	Minimum Air Temperature [K]
Float	135	-	Maximum Air Temperature [K]
Float	136	-	Dew Point Temperature [K]
Float	137	-	Wind Speed [m/s]
Float	138	-	Cloudiness
Float	139	-	Sun Zenith
Float	140	-	Sun Azimuth
Float	141	-	Instant Power [w]
Float	142	-	Current Hour Energy Consumption [J]
Float	143	-	Current Day Energy Consumption [J]
Float	144	-	Current Week Energy Consumption [J]
Float	145	-	Current Month Energy Consumption [J]
Float	146	-	Last Hour Energy Consumption [J]
Float	147	-	Last Day Energy Consumption [J]
Float	148	-	Last Week Energy Consumption [J]
Float	149	-	Last Month Energy Consumption [J]
Float	150	A	Temperature [K]

Float	151	B	Temperature [K]
Float	152	C	Temperature [K]
Float	153	D	Temperature [K]
Float	154	E	Temperature [K]
Float	155	F	Temperature [K]
Float	156	G	Temperature [K]
Float	157	H	Temperature [K]
Float	158	I	Temperature [K]
Float	159	J	Temperature [K]
Float	160	K	Temperature [K]
Float	161	L	Temperature [K]
Float	162	M	Temperature [K]
Float	163	N	Temperature [K]
DateTime	65	-	Date And Time

ANEXO D: Función de ejemplo en Matlab para la lectura de variables de Home I/O

```

function read(block)
% Level-2 MATLAB file S-Function for unit delay demo.
% Copyright 1990-2009 The MathWorks, Inc.

    setup(block);

%endfunction

function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 0;
    block.NumOutputPorts = 6;

    %% Setup functional port properties to dynamically
    block.SetPreCompPortInfoToDefaults;

    block.OutputPort(1).Dimensions = 1;

    %% Set block sample time to [0.1 0]
    block.SampleTimes = [0.02 0];

    %% Set the block simStateCompliance to default (i.e., same as a
    built-in block)
    block.SimStateCompliance = 'DefaultSimState';
    path = strcat(pwd, '\EngineIO.dll');
    NET.addAssembly(path);
    %% Register methods
    block.RegBlockMethod('Outputs', @Output);
%endfunction

function Output(block)
EngineIO.MemoryMap.Instance.Update();

```

```

%Temperatura cuarto niños
block.OutputPort(1).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(58,
EngineIO.MemoryType.Input).Value);
%Luminosidad cuarto niños
block.OutputPort(2).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(57,
EngineIO.MemoryType.Input).Value);
%Nivel de apertura persianas cuarto niños
block.OutputPort(3).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(60,
EngineIO.MemoryType.Input).Value);
%DETECTOR DE PUERTA 1 cuarto niños
block.OutputPort(4).Data =
double(EngineIO.MemoryMap.Instance.GetBit(136,
EngineIO.MemoryType.Input).Value);
%DETECTOR DE PUERTA 2 cuarto niños
block.OutputPort(5).Data =
double(EngineIO.MemoryMap.Instance.GetBit(137,
EngineIO.MemoryType.Input).Value);
%Potencia instantanea consumida
block.OutputPort(6).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(141,
EngineIO.MemoryType.Memory).Value);

%endfunction

```

ANEXO E: Función de ejemplo en Matlab para la escritura de variables de Home I/O

```

function write(block)
% Level-2 MATLAB file S-Function for unit delay demo.
% Copyright 1990-2009 The MathWorks, Inc.

    setup(block);

%endfunction

function setup(block)
%% Register number of input and output ports
block.NumInputPorts = 4;
block.NumOutputPorts = 0;

%% Setup functional port properties to dynamically
block.SetPreCompPortInfoToDefaults;

block.InputPort(1).Dimensions = 1;

%% Set block sample time to [0.1 0]
block.SampleTimes = [0.02 0];

%% Set the block simStateCompliance to default (i.e., same as a
built-in block)
block.SimStateCompliance = 'DefaultSimState';

path = strcat(pwd, '\EngineIO.dll');

```

```

NET.addAssembly(path);

%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction

function Output(block)
    %Calentador niños
    Heater = EngineIO.MemoryMap.Instance.GetFloat(79,
EngineIO.MemoryType.Output);
    Heater.Value = block.InputPort(1).Data;
    %Luces niños
    Luz = EngineIO.MemoryMap.Instance.GetFloat(78,
EngineIO.MemoryType.Output);
    Luz.Value = block.InputPort(2).Data;
    %PersianasSubida
    PersianasUP = EngineIO.MemoryMap.Instance.GetBit(98,
EngineIO.MemoryType.Output);
    PersianasUP.Value = block.InputPort(3).Data;
    %PersianasBajada
    PersianasDOWN = EngineIO.MemoryMap.Instance.GetBit(99,
EngineIO.MemoryType.Output);
    PersianasDOWN.Value = block.InputPort(4).Data;

    EngineIO.MemoryMap.Instance.Update();

%endfunction

```

ANEXO F: Función de lectura de las temperaturas de todas las habitaciones equipadas con calefactor de la vivienda en Home I/O

```

function read(block)
% Level-2 MATLAB file S-Function for unit delay demo.
% Copyright 1990-2009 The MathWorks, Inc.

    setup(block);

%endfunction

function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 0;
    block.NumOutputPorts = 12;

    %% Setup functional port properties to dynamically
    block.SetPreCompPortInfoToDefaults;

    block.OutputPort(1).Dimensions = 1;

    %% Set block sample time to [0.1 0]
    block.SampleTimes = [0.1 0];

    %% Set the block simStateCompliance to default (i.e., same as a
    built-in block)

```

```

block.SimStateCompliance = 'DefaultSimState';
path = strcat(pwd, '\EngineIO.dll');
NET.addAssembly(path);
% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction

```

```

function Output(block)
EngineIO.MemoryMap.Instance.Update();

%Temperatura zona A
block.OutputPort(1).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(1,
EngineIO.MemoryType.Input).Value);
%Temperatura zona D
block.OutputPort(2).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(13,
EngineIO.MemoryType.Input).Value);
%Temperatura zona E
block.OutputPort(3).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(25,
EngineIO.MemoryType.Input).Value);
%Temperatura zona G
block.OutputPort(4).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(46,
EngineIO.MemoryType.Input).Value);
%Temperatura zona H
block.OutputPort(5).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(58,
EngineIO.MemoryType.Input).Value);
%Temperatura zona I
block.OutputPort(6).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(69,
EngineIO.MemoryType.Input).Value);
%Temperatura zona J
block.OutputPort(7).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(81,
EngineIO.MemoryType.Input).Value);
%Temperatura zona K
block.OutputPort(8).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(92,
EngineIO.MemoryType.Input).Value);
%Temperatura zona L
block.OutputPort(9).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(104,
EngineIO.MemoryType.Input).Value);
%Temperatura zona M
block.OutputPort(10).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(116,
EngineIO.MemoryType.Input).Value);
%Temperatura zona N
block.OutputPort(11).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(128,
EngineIO.MemoryType.Input).Value);
%Potencia instantanea consumida

```

```

block.OutputPort(12).Data =
double(EngineIO.MemoryMap.Instance.GetFloat(141,
EngineIO.MemoryType.Memory).Value);
%endfunction

```

ANEXO G: Función de apagado o modulación automática de los calefactores de la vivienda

```

function modulacion_y_apagado(block)
% Level-2 MATLAB file S-Function for unit delay demo.
% Copyright 1990-2009 The MathWorks, Inc.

    setup(block);

%endfunction

function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 12;
    block.NumOutputPorts = 11;

    %% Setup functional port properties to dynamically
    block.SetPreCompPortInfoToDefaults;

    block.InputPort(1).Dimensions = 1;
    block.OutputPort(1).Dimensions = 1;

    %% Set block sample time to [0.1 0]
    block.SampleTimes = [0.1 0];

    %% Set the block simStateCompliance to default (i.e., same as a
    built-in block)
    block.SimStateCompliance = 'DefaultSimState';

    %% Register methods
    block.RegBlockMethod('Outputs', @Output);
%endfunction

function Output(block)
%% lectura entradas
Diferencia=block.InputPort(1).Data;
EstadoA=double(block.InputPort(2).Data);
EstadoD=block.InputPort(3).Data;
EstadoE=block.InputPort(4).Data;
EstadoG=block.InputPort(5).Data;
EstadoH=block.InputPort(6).Data;
EstadoI=block.InputPort(7).Data;
EstadoJ=block.InputPort(8).Data;
EstadoK=block.InputPort(9).Data;
EstadoL=block.InputPort(10).Data;
EstadoM=block.InputPort(11).Data;
EstadoN=block.InputPort(12).Data;
%%
global ReduccionA

```

```

global ReduccionD
global ReduccionE
global ReduccionG
global ReduccionH
global ReduccionI
global ReduccionJ
global ReduccionK
global ReduccionL
global ReduccionM
global ReduccionN
if (Diferencia>0)
    if(ReduccionM<10 && EstadoM>0)
        ReduccionM=ReduccionM+2;
    elseif(ReduccionA<10 && EstadoA>0)
        ReduccionA=ReduccionA+2;
    elseif(ReduccionE<10 && EstadoE>0)
        ReduccionE=ReduccionE+2;
    elseif(ReduccionG<10 && EstadoG>0)
        ReduccionG=ReduccionG+2;
    elseif(ReduccionD<10 && EstadoD>0)
        ReduccionD=ReduccionD+2;
    elseif(ReduccionN<10 && EstadoN>0)
        ReduccionN=ReduccionN+2;
    elseif(ReduccionH<10 && EstadoH>0)
        ReduccionH=ReduccionH+2;
    elseif(ReduccionJ<10 && EstadoJ>0)
        ReduccionJ=ReduccionJ+2;
    elseif(ReduccionL<10 && EstadoL>0)
        ReduccionL=ReduccionL+2;
    elseif(ReduccionI<10 && EstadoI>0)
        ReduccionI=ReduccionI+2;
    elseif(ReduccionK<10 && EstadoK>0)
        ReduccionK=ReduccionK+2;
    end
elseif (Diferencia<-300)
    if(ReduccionK>0 && EstadoK>0)
        ReduccionK=ReduccionK-2;
    elseif(ReduccionI>0 && EstadoI>0)
        ReduccionI=ReduccionI-2;
    elseif(ReduccionL>0 && EstadoL>0)
        ReduccionL=ReduccionL-2;
    elseif(ReduccionJ>0 && EstadoJ>0)
        ReduccionJ=ReduccionJ-2;
    elseif(ReduccionH>0 && EstadoH>0)
        ReduccionH=ReduccionH-2;
    elseif(ReduccionN>0 && EstadoN>0)
        ReduccionN=ReduccionN-2;
    elseif(ReduccionD>0 && EstadoD>0)
        ReduccionD=ReduccionD-2;
    elseif(ReduccionG>0 && EstadoG>0)
        ReduccionG=ReduccionG-2;
    elseif(ReduccionE>0 && EstadoE>0)
        ReduccionE=ReduccionE-2;
    elseif(ReduccionA>0 && EstadoA>0)
        ReduccionA=ReduccionA-2;
    elseif(ReduccionM>0 && EstadoM>0)
        ReduccionM=ReduccionM-2;

```

```

        end
    end
    %% Escritura salidas
    %Reduccion zona A
    block.OutputPort(1).Data = double(ReduccionA);
    %Reduccion zona D
    block.OutputPort(2).Data = double(ReduccionD);
    %Reduccion zona E
    block.OutputPort(3).Data = double(ReduccionE);
    %Reduccion zona G
    block.OutputPort(4).Data = double(ReduccionG);
    %Reduccion zona H
    block.OutputPort(5).Data = double(ReduccionH);
    %Reduccion zona I
    block.OutputPort(6).Data = double(ReduccionI);
    %Reduccion zona J
    block.OutputPort(7).Data = double(ReduccionJ);
    %Reduccion zona K
    block.OutputPort(8).Data = double(ReduccionK);
    %Reduccion zona L
    block.OutputPort(9).Data = double(ReduccionL);
    %Reduccion zona M
    block.OutputPort(10).Data = double(ReduccionM);
    %Reduccion zona N
    block.OutputPort(11).Data = double(ReduccionN);

    %endfunction

```

ANEXO H: Función de escritura de las tensiones de entrada a los calefactores calefactor de cada habitación de la vivienda en Home I/O

```

function write(block)
% Level-2 MATLAB file S-Function for unit delay demo.
% Copyright 1990-2009 The MathWorks, Inc.

    setup(block);

%endfunction

function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 12;
    block.NumOutputPorts = 0;

    %% Setup functional port properties to dynamically
    block.SetPreCompPortInfoToDefaults;

    block.InputPort(1).Dimensions = 1;

    %% Set block sample time to [0.1 0]
    block.SampleTimes = [0.1 0];

    %% Set the block simStateCompliance to default (i.e., same as a
    built-in block)
    block.SimStateCompliance = 'DefaultSimState';

```



```

path = strcat(pwd, '\EngineIO.dll');
NET.addAssembly(path);

%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction

function Output(block)
    %Calefaccion zona A
    HeaterA = EngineIO.MemoryMap.Instance.GetFloat(1,
EngineIO.MemoryType.Output);
    HeaterA.Value = block.InputPort(1).Data;
    %Calefaccion zona D
    HeaterD = EngineIO.MemoryMap.Instance.GetFloat(34,
EngineIO.MemoryType.Output);
    HeaterD.Value = block.InputPort(2).Data;
    %Calefaccion zona E
    HeaterE = EngineIO.MemoryMap.Instance.GetFloat(45,
EngineIO.MemoryType.Output);
    HeaterE.Value = block.InputPort(3).Data;
    %Calefaccion zona G1
    HeaterE = EngineIO.MemoryMap.Instance.GetFloat(67,
EngineIO.MemoryType.Output);
    HeaterE.Value = block.InputPort(4).Data;
    %Calefaccion zona G2
    HeaterE = EngineIO.MemoryMap.Instance.GetFloat(68,
EngineIO.MemoryType.Output);
    HeaterE.Value = block.InputPort(5).Data;
    %Calefaccion zona H
    HeaterH = EngineIO.MemoryMap.Instance.GetFloat(79,
EngineIO.MemoryType.Output);
    HeaterH.Value = block.InputPort(6).Data;
    %Calefaccion zona I
    HeaterI = EngineIO.MemoryMap.Instance.GetFloat(91,
EngineIO.MemoryType.Output);
    HeaterI.Value = block.InputPort(7).Data;
    %Calefaccion zona J
    HeaterJ = EngineIO.MemoryMap.Instance.GetFloat(102,
EngineIO.MemoryType.Output);
    HeaterJ.Value = block.InputPort(8).Data;
    %Calefaccion zona K
    HeaterK = EngineIO.MemoryMap.Instance.GetFloat(113,
EngineIO.MemoryType.Output);
    HeaterK.Value = block.InputPort(9).Data;
    %Calefaccion zona L
    HeaterL = EngineIO.MemoryMap.Instance.GetFloat(124,
EngineIO.MemoryType.Output);
    HeaterL.Value = block.InputPort(10).Data;
    %Calefaccion zona M
    HeaterM = EngineIO.MemoryMap.Instance.GetFloat(135,
EngineIO.MemoryType.Output);
    HeaterM.Value = block.InputPort(11).Data;
    %%Calefaccion zona N
    HeaterN = EngineIO.MemoryMap.Instance.GetFloat(148,
EngineIO.MemoryType.Output);

```

```
HeaterN.Value = block.InputPort(12).Data;  
EngineIO.MemoryMap.Instance.Update();  
%endfunction
```

